InfoBurst Enterprise

Command Shell (IBShell)

Reference Guide

Version 2012.8 b120 - Rev date August 9th 2012

Table of Contents

Introduction

An InfoBurst system contains a number of objects; these can be system objects such as Platform, User, Group, Folder or user objects such as Documents, Bursts etc.

An object can be created in a number of ways:

- Via the user interface
- Using the IBShell “new” and “add” commands
- Via the InfoBurst API’s

Software developers will use the SOAP/Ajax API’s to integrate functions of the InfoBurst platform within their own solutions.

For non developers, the **IBShell** utility provides a command-line interface to many of the available features and supports a script mode whereby commands can be stored in an external file which allows tasks to be automated / repeated.

To learn more about how to build custom solutions using the API’s, please refer to the Web Service API Reference.

Using the IBShell utility

To start the tool, run **IBShell.exe** from the InfoBurst application folder (typically c:\program files\infosol\infoburst2009).

If you wish to have a local copy on your workstation, you must have the .Net Framework 3.5 installed (and any prerequisites such as .Net 2.0 and .Net 3.5). You can run a client only install of InfoBurst to install only the minimum set of files required to communicate with the server and connect to the repository.

The command line tool will issue a prompt and the first task is to get connected to an InfoBurst server.

Use the “connect” command followed by the name of the server (or just use connect with no parameters to connect to the local server).

Once connected, you will need to obtain a logon token to authenticate a user with the system and establish how long the token will last before expiring.

Here is an example command line session:

```
IBShell.exe
InfoBurst Enterprise Shell Utility, (c) InfoSol 2008-2011
> connect
. Connecting to InfoBurst Enterprise on server MYSERVER
* Connected OK, version is 2011.10.117 - Enterprise Edition
* Licensed to xxxx
> get token user="admin" password="xxx"
* OK
> status
  Connected to 'MYSERVER'
  Token          : c1f951ac-336a-4fcd-836f-c85d0f1be1aa
    Issued       : 7/18/2010 12:06:38 PM
    User         : ADMIN
    UserID       : 1
    RootFolder   : \users\Admin\
    RootFolderID : 8
    LastUsed     : 7/18/2010 12:06:38 PM
    Expires      : 7/18/2010 12:11:38 PM
    UsageCtr     : 0
    Client Addr  : 192.168.100.99
    Client Port  : 50613
    Client Info  : IBShell
>
```

The "connect" command was used, followed by the "get token" command and finally the "status command".

To get a list of functions, use the "?" or "help" command.

Using a script

You can create a text file with a list of commands and specify the name of the script file when starting the command line tool.

This can be useful to automatically connect and get a logon token to save having to manually execute the same commands each time.

It is also useful to be able to write and invoke scripts to help with system management and testing.

Here is an example script called “cmd.ibs” that contains the connect and get token commands:

```
Connect
Get token user=”john” password=”mypassword”
```

To use the script, specify the name/path as below:

IBShell.exe **cmd.ibs**

IBShell Command Reference

If a command has parameters, use **Xml style syntax** such as

Parameter = “value”

All parameters must be enclosed by double quotes however they will be added automatically so you can say Parameter=10 and this will be treated as Parameter=”10”.

If a command has a single parameter, you can omit the parameter name.

If a command has multiple parameters, you can omit the first parameter name but must supply all other parameters using the Parameter = “value” syntax.

If a parameter contains spaces and you are using the single parameter syntax without the parameter name, be sure to enclose the value with quotes such as “Sales Reports”.

Most commands have an abbreviated name; these will be shown in brackets.

For example, the “connect” command has an optional parameter called “Server” and two abbreviations “c” and “conn”.

The following are the same:

- C myserver
- Conn server=”myserver”
- Connect “myserver”

As the connect command only has one parameter, you do not need to specify “Server=” as it will be assumed.

The “Get Token” command has more than one parameter therefore the following are the same:

- Get Token User=”john” password=”mypassword”
- Get Token john password=”mypassword”
- Get Token “john smith” password=”zyx123”
- Get Token User=”john smith” password=”zyx123”

Connecting, Disconnecting and Token Management

These commands establish a connection to an InfoBurst system and manage logon tokens.

Connect (conn, c)

Connect to an InfoBurst system.

Parameters:

Server	optional	Name or address of InfoBurst Server
Port	optional	Port number if not 8550
SSL	optional	Use secure connection (https://)

Examples:

```
1    Connect
2    Connect ibppserver
3    Conn server="ibppserver"
4    C server="ibppserver:9050"
5    C port="9050"
6    C server="ibeprod.infosol.com" ssl=true
```

Example 1 connects to the local server.

Examples 2 and 3 connect to a named server.

Example 4 connects to a named server on a non-default port.

Example 5 connects to the local server on a non-default port.

Example 6 connects to a server using https://

Disconnect (disc)

Disconnect from the InfoBurst system.

Get (g) Token (tkn)

Get a logon token for a given user.

The Expiry (#Minutes) sets the number of minutes the token will be kept alive on the server after the last use. If the token is re-used before the expiry, the expiry time is reset.

Parameters:

User	required	Name of InfoBurst User
Password	required	Password for user
Expiry	optional (5)	Number of minutes to keep after idle.

Examples:

```
1    Get Token user="john" password=" mypassword"

2    G Tkn user="john" password="mypassword" expiry="10"
```

Example 1 gets a logon token with default expiry of 5 minutes after idle.

Example 2 gets a logon token with an expiry of 10 minutes after idle.

Use (u) Token (tkn)

Uses a previous token obtained for a given user.

Parameters:

User	required	Name of InfoBurst User

Examples:

```
1    Use Token user="john"

2    U Tkn john
```

Release (r, rel) Token (tkn)

Release a token obtained for a given user. This will close any resources/sessions that were used.

Parameters:

User	required	Name of InfoBurst User

Examples:

```
1    Release Token user="john"

2    R Tkn john
```

List (l,lst) Tokens

Display all tokens that have been created during this IBShell session.

Examples:

```
1    List Tokens

2    L Tokens
```

List (l,lst) All Tokens

Display all tokens that have been created on the InfoBurst server. **Requires Admin role**.

Parameters:

Sort	optional	**user**/issued/expiry/usage

Examples:

```
1    List All Tokens

2    L All Tokens sort="expiry"
```

Example 1 lists all issues tokens sorted by user.

Example 2 lists all tokens sorted by expiry time.

Miscellaneous commands

Exit (x,quit,q)

Stop the IBShell session. If you are connected an automatic disconnect will be issued.

Status (stat,s)

Display the current status of the IBShell session. Details about the logon token will be displayed.

SysInfo(info)

Display general system information about InfoBurst including version number and build.

Using

Assigns default values to parameters. Very useful when working with objects such as Bursts as the current Burst and Document can be set once and then subsequent commands can omit the defaults.

Most commands will assign values automatically; for example when using the New or Replace commands on objects that support additional commands an automatic USING statement is implied.

Multiple settings are supported in a single command.

To see all current values, simply say "using".

Example 1 (no using):

```
New Burst Name="MyBurst"

Add Document Burst="MyBurst" DocID="Sales per Region"

Set Parameter Burst="MyBurst" Document="Sale per Region" State="AZ"
```

Example 2 (with using):

```
New Burst Name="MyBurst"

Add Document DocID="Sales per Region"

Set Parameter State="AZ"
```

Clear Using

Clears all using values.

To Clear a specific value (or values) give a unique reference (such as u1) and the name(s) of the Using variables to clear

Example:

```
Clear Using

Clear Using u1="Burst" u2="DocID"
```

Var

Assigns values to variables. Very useful when working with scripts that need to be customized for certain users etc.

A Variable is used by prefixing the name with a "**$**" (see examples below).

Multiple settings are supported in a single command.

To see all current values, simply say "var".

The "$" replacement character can be changed by using Set Var char="#" (see below).

Example:

```
Var user="John" email="jwilcox@infoburst.com"

New User name="$user" password="x" email="$email"

New Folder name="Test" parent="\users\$user"
```

Clear Var

Clears all Variables.

To Clear a specific variable (or variables) give a unique reference (such as v1) and the name(s) of the Variables to clear

Example:

```
Clear Var

Clear Var v1="user"
```

Set Var check="true" or "false"

Will check that variable substitution does not result in any unassigned values. By default this is "false" as there is a possibility that the script will issue a command that you do not wish to be interpreted as a variable (for example an InfoBurst delivery using a macro).

Set Var char="$"

Sets the variable replacement character. By default this is "$" but if you wish to check variables and also wish to use InfoBurst delivery macros that use the same character (e.g. [$Burst]) then you can change the char to a unique value such as "#".

Exec Script

Execute the contents of an external script file.

If the file cannot be found in the current folder, an attempt will be made to locate the file in the current "Scripts" folder (if present).

If no extension is given, ".ibs" will be added.

If you wish to execute multiple scripts in a series but only execute a script if the previous script was successful, use the **ResultVar** to assign the script result and then specify the variable as the **StartIf** value on the subsequent command.

Parameters:

Name	required	Name/Path of script file
ResultVar	optional	Name of Variable to store result (0/1)
StartIf	optional	Name of Variable to test ="1"

Examples:

```
Exec Script name="c:\scripts\users.ibs"

Exec Script name="c:\scripts\users.ibs" ResultVar="UsersOK"

Exec Script name="c:\scripts\groups.ibs" StartIf="UsersOK"
```

Pause

Pause the current Script for a # seconds. Can be used to stop a Script while waiting for the system to process tasks (such as when Cataloging documents).

Parameters:

Seconds	required	# Seconds to wait

Examples:

```
Pause Seconds="10"
```

Stop

Request that the InfoBurst system service shuts down.

Parameters:

Minutes	required	How many minutes from now (5-240)
When	optional	use “NOW” for immediate stop

Examples:

```
Stop Minutes=”10”
```

Home

Return to the current user’s home folder.

Working with the Folder system

These commands allow you to display the objects contained in folders and also to create new folders or generally manage objects within folders.

Dir

List objects in a folder. After obtaining a logon token, the current folder will be set to the "user root". This command does not change the current folder (see CD).

If a Folder is specified, use "\Folder" to start from the root or the name will be considered relative to the current folder.

Parameters:

Folder	optional	Name of Folder

Examples:

```
1    Dir

2    Dir "\Sales Reports"

3    Dir Folder="MyReports"
```

Example 1 lists objects in the current folder. Use the "CD" command to display the current folder.

Example 2 lists objects in the specified root folder.

Example 3 lists objects in the relative folder "MyReports". If the current folder was "\Sales" the path would be "\Sales\MyReports".

CD

Display current folder or change current folder.

Parameters:

Folder	optional	Name of Folder

Examples:

```
1    CD

2    CD "\Sales Reports"

3    CD Folder="MyReports"
```

Example 1 displays the current folder.

Example 2 changes the current folder to the specified root folder.

Example 3 changes the current folder to the relative folder "MyReports". If the current folder was "\Sales" the path would be "\Sales\MyReports".

MkDir (md,newdir)

Create a new sub-folder. First ensure that the current folder has been set to the correct "parent" folder.

Parameters:

Folder	required	Name of Folder

Examples:

```
1    MkDir "Sales Reports"

1    MkDir Folder="Sales Reports"
```

Rename(ren)

Change the name of an existing object or folder.

To specify a Folder object, use Object="\Folder\" or use the Rename Folder command.

To rename a User, use the command **Rename User** Name="x" NewName="y"

To rename a Group, use the command **Rename Group** Name="x" NewName="y"

To rename a Platform, use the command **Rename Platform** Name="x" NewName="y"

To rename a Folder, use the command **Rename Folder** Name="x" Folder="TargetFolder"

Parameters:

Object	required	Name or Path of Object
NewName	required	New name for Object

Examples:

```
1    Rename object="\Sales\ReportA" newname="ReportB"
2    Ren folder "\Sales" newname="2007 Sales"
3    Rename group "Sales" newname="SalesTesmA"
```

Example 1 renames an object "ReportA" in folder "\Sales" to be called "ReportB".

Example 2 renames a folder "\Sales" to be called "2007 Sales".

Example 3 renames a group.

Copy(cpy)

Copy an object either within the same folder or to a different folder, or copy a folder (and its contents) to another folder.

If NewName is blank and no folder is given, the new object will be prefixed “Copy Of”.

To specify a Folder object, use Object=”\Folder\” or use the Copy Folder command.

To copy a User, use the command **Copy User** Name=”x” NewName=”y”

To copy a Group, use the command **Copy Group** Name=”x” NewName=”y”

To copy a Platform, use the command **Copy Platform** Name=”x” NewName=”y”

To copy a Folder, use the command **Copy Folder** Name=”x” Folder=”TargetFolder”

Parameters:

Object	required	Name or Path of Object
Folder	optional	Destination Folder
NewName	optional	New name for Object

Examples:

```
1  Copy object=”\Sales\ReportA”
2  Copy ”\Sales\ReportA” name=”ReportB”
3  Copy object=”\Sales\ReportA” folder=”\Sales07”
4  Copy object=”\Sales\Q1\” folder=”\Sales\Archive”
5  Copy User name=”john” newname=”steve”
6  Copy Folder name=”Sales\Q1” folder=”\Sales\Archive”
```

Example 1 copies an object “ReportA” in folder “\Sales” to be called “Copy Of ReportA”.

Example 2 copies an object “ReportA” in folder “\Sales” to be called “ReportB”.

Example 3 copies an object “ReportA” in folder “\Sales” to folder “\Sales07”.

Example 4 copies an entire folder “\Sales\Q1” to folder “\Sales\Archive”.

Example 5 copies a user.

Example 6 is the same as 4 but uses the Copy Folder syntax (note does not need trailing ****)

Move(mv)

Move an object to a different folder, or move a folder (and its contents) to another folder.

To specify a Folder object, use Object="\Folder\" or use the Move Folder command.

To move a Folder, use the command **Move Folder** Name="x" Folder="TargetFolder"

Parameters:

Object	required	Name or Path of Object
Folder	required	Destination Folder

Examples:

```
1  Move object="\Sales\ReportA" folder="\Sales07"

2  Move object="\Sales\Q1\" folder="\Sales\Archive"

3  Move Folder name="\Sales\Q1" folder="\Sales\Archive"
```

Example 1 moves an object "ReportA" in folder "\Sales" to folder "\Sales07".

Example 2 moves an entire folder "\Sales\Q1" to folder "\Sales\Archive".

Example 3 is the same as 2 but uses the Move Folder syntax (note does not need trailing ****)

Delete(del)

Delete a system object, user object or an entire folder (and its contents).

This command can also be used to delete 'sub items' from an existing Object (such as User Logon, Group Member, Database Qry etc).

An Object can be deleted using either it's name or ID. If using ID, multiple ID values can be given (e.g. ID="10,11,15").

If a folder contains objects (or other sub-folders), it cannot be deleted until these have also been deleted.

Parameters:

Object-Type	required	Type of Object
Name	optional	Name or Path of Object
ID	optional	Unique ID for Object

The values for Object-Type are:

- User
- Group
- Platform
- Folder
- Document
- Burst
- Schedule
- Event
- DBConnection
- LdapConnection
- Text
- List
- FileList
- LdapList

- XIList
- MultiColumnList

Examples:

```
1  Delete document name="\Sales\ReportA"
2  Delete folder name="\Sales\Q1"
3  Delete User "Jane"
4  Delete dbconnection id="123"
5  Delete Document id="10,15,18"
```

Example 1 deletes an object "ReportA" in folder "\Sales".

Example 2 deletes an entire folder "\Sales\Q1".

Example 3 deletes the system object of type User with name="Jane".

Example 4 deletes a database connection object with an ID of 123.

Example 5 deletes 3 documents with the given ID values.

Deleting a User Logon

Delete logon user="name" platform="platform-name"

Deleting a member from a Group

Delete groupmember group="group" user="user"

Deleting a Database Query

Delete query dbconn="name" name="query-name"

Deleting a Column from a MultiColumnList

Delete column mclist="list" name="column-name"

Deleting a Row from a MultiColumnList

Delete row mclist="list" number="row-number"

Display(d,disp)

Display detailed information about any Object.

To display a system object, use the following syntax:

- **Display User** "user" or **Display object="\Users**user"
- **Display Group** "group" or **Display object="\Groups**group"
- **Display Platform** "platform" **or Display object="\Platforms**platform"

Parameters:

Object	required	Name or Path of Object

Examples:

```
1  Display object="\Sales\ReportA"
2  Display User "Jane"
3  Display SalesReport
4  Display "My Burst"
```

Example 1 displays an object "ReportA" in folder "\Sales".

Example 2 displays the system object of type User with name="Jane".

Example 3 displays the Object "SalesReport" in the current Folder.

Example 4 displays the Object "My Burst" in the current Folder.

Owner

Display or change the Owner of a User Object.

Parameters:

Object	required	ID or Name of Object
New	optional	New Owner ID Name or ID

Refs

Display references to a given User Object.

Parameters:

Object	required	ID or Name of Object
Query	optional	ID or Name of Query

Links

Display links from a given User Object.

Parameters:

Object	required	ID or Name of Object

List (l,lst)

The List command is similar to the Dir command when used to list folders but supports other options to list system objects, system requests and system logs

System Servers

- List Servers
    - Lists all servers that are configured with this InfoBurst system

System Objects

- List Platforms
    - Lists all document platforms
- List Users
    - Lists all Users
- List Groups
    - Lists all Groups

System Parameters and Aliases

- List Parameters
    - Lists all unique document parameters (from all cataloged documents)
- List Alias
    - Lists all user defined Aliases (use Display Alias to see details)

System Paths and Printers

- List Paths
    - Lists all registered UNC Paths
- List Printers
    - Lists all user registered Printers
- List Printers installed=”true”
    - Lists all installed Printers on the InfoBurst server

System MetaTags

- List Tags
    - Lists all registered Object MetaTags

System Configuration

- List Config
    - Lists all system configuration values

Folders

- List Folders
    - Lists all root folders in the InfoBurst repository
- List Folder=”folder” Objects=”**ObjectFilter**”
    - Lists objects in a specific folder in the InfoBurst repository

System Requests

- List Requests
    - Lists all system requests (such as Cataloging etc)

System Activity

- List Activity
    - Lists all system activity for each system/platform Queue

System Schedules

- List Schedules [select=”n”]
    - Lists all pending Schedules and optionally select one for use with hold/force/stop
- List Schedules **date=”futureDate”**
    - Lists all future Schedules for a given date
- List Schedules **type=”running”** (or type=”exec”) [select=”n”]
    - Lists all running Schedules and optionally select one for use with hold/force/stop
- List Schedules **type=”finished”** (or type=”done”) [select=”n”]

- Lists all finished Schedules

System Events

- List Events
  - Lists all pending Events

System Audit Log

- List Audit
  - Lists all Audit records for all users for today
- List Audit **id**=”10” **from**=”11/1/08” **to**=”11/2/08”
  - Lists all Audit records for a specific user for a date range

System Document Log

- List Document
  - Lists all Document processing records for all Documents for today
- List Document **id**=”10” **from**=”11/1/08” **to**=”11/2/08”
  - Lists all Document processing records for a specific Document for a date range

System Delivery Log

- List Delivery
  - Lists all Document delivery records for all Documents for today
- List Delivery **id**=”10” **from**=”11/1/08” **to**=”11/2/08”
  - Lists all Document processing records for a specific Document for a date range

Logon Tokens

- List Tokens
  - Lists all tokens used in the IBShell session
- List All Tokens
  - Lists all tokens that exist on the InfoBurst server

Burst's on the 'Alert' list

- List Alert
    - Lists all Burst's that have been added to the system 'alert' list

Examples:

```
1  List folder="\Sales"

2  List folder="\Sales" objects="Doc"

3  List folder="\Sales" objects="Doc(CR)"

4  List folders
```

Example 1 lists all objects in the folder "\Sales".

Example 2 lists only document objects in the folder "\Sales".

Example 3 lists only crystal reports in the folder "\Sales".

Example 4 lists all root folders in the repository.

PList (pl)

This command lists folders, documents, users, groups & categories for a given platform.

The available types are:

- Folders
- SubFolders
- Documents
- Users
- Groups
- Categories

The default Type is “folders”

Parameters:

Platform	required	ID or Name for Platform
Type	optional	Type of item to list
Folder	optional	Folder if required
Objects (obj,o)	optional	**ObjectFilter**

Examples:

```
1  PList platform=”xir2”
2  PList platform=”xir2” type=”subfolders” folder=”Sales”
3  PList platform=”xir2” type=”documents” folder=”Sales”
4  Pl platform=”xir2” type=”users”
```

Find(f)

Use this command to find objects in the repository based on either object type and optional property condition or MetaData Tags.

A property condition is a condition that is object specific, for example the Document object has properties such as Size that contain the size in bytes of the cataloged file.

The Find command is not based on Folders and therefore can find objects across the whole repository.

Parameters:

Objects (obj,o)	optional	**ObjectFilter**
Tags	optional	Tag name(s)
Name	optional	Object name filter
Condition	optional	Property Condition
Sort	optional	**name**/created/changed

Examples:

```
1  Find objects="documents"
2  Find obj="doc(CR)" condition="size>25000"
3  Find tags="bo"
4  Find o="burst" name="Sales*"
```

Example 1 finds all documents.

Example 2 finds all Crystal Reports documents with a size > 25000 bytes.

Example 3 finds all objects tagged with "bo".

Example 4 finds all burst objects with names starting "Sales".

Creating and managing System Objects

These commands allow you to create and manage platforms, users, groups and folders.

New Platform

Creates a new document Platform.

When a new platform is created, the next step is normally to add logons to the new platform for selected users (see **add logon**).

Parameters:

Type	required	**PlatformType**
Name	required	Name of Platform
Server	optional	Server / CMC Name
Temp	optional	Folder for temporary objects
Auth	required	**PlatformAuthType**
AppServer	optional	App Server
AppServerType	optional	**PlatformAppServerTechnology**
AppServerPort(Port)	optional	App Server Port # (80)
AppServerPublicURL(URL)	optional	Public address for App Server
ConnType	optional	**PlatformConnectionType**
Timeout	optional	# Mins to allow for a request
Preview	optional	Dflt "True". Create preview images

Example:

```
New Platform type="XIR2" name="XIProd" Server="ibtest2k" auth="E"
AppServerType="java" Port="8080"
```

Modify Platform

Modifies an existing Platform. Only the name and authentication can be changed.

Parameters:

Name	required	Name of Platform
NewName	optional	New name of Platform
Temp	optional	Folder for temporary objects
Auth	optional	**PlatformAuthType**
AppServer	optional	App Server
AppServerType	optional	**PlatformAppServerTechnology**
AppServerPort(Port)	optional	App Server Port # (80)
AppServerPublicURL(URL)	optional	Public address for App Server
ConnType	optional	**PlatformConnectionType**
Timeout	optional	# Mins to allow for a request
Preview	optional	Create preview images

Example:

```
Modify Platform name="XIProd" newname="ibtest2k"
```

Modify Queue

Modifies an existing Platform Queue.

Parameters:

Platform	required	Name of Platform
Name	required	Name of Queue
Size	required	New Max # Jobs

Example:

```
Modify Queue Platform="XIProd" name="WI" Size=15
```

The range of allowed Size is as follows:

For Platform of type **DI20** or **DI30**

Queue "BO"	MaxJobs	1-5
Queue "BO_CATALOG"	MaxJobs	1-2

For Platform of type **XI20** or **XI30**

Queue "CR"	MaxJobs	1-10
Queue "CR_CATALOG"	MaxJobs	1-2
Queue "WI"	MaxJobs	1-20
Queue "WI_CATALOG"	MaxJobs	1-5
Queue "DI"	MaxJobs	1-10
Queue "DI_CATALOG"	MaxJobs	1-2
Queue "DI_FILTER"	MaxJobs	1-5

Delete Platform

Deletes an existing Platform.

If documents have been cataloged from this platform the deletion will fail.

Parameters:

Name	required	Name of Platform

Example:

```
Delete Platform name="XIProd"
```

New Folder

Creates a new Folder where objects can be stored.

If you wish to change the name of the Folder at a later time, use the **RENAME** command.

Parameters:

Name	required	Name of Folder
Parent	required	Parent Folder

Example:

```
New Folder Name="Sales Reports" Parent="\"

New Folder Name="DivA" Parent="\Sales Reports"

New Folder Name="Weekly" Parent="\Sales Reports\DivA"
```

Delete Folder

Deletes an existing Folder and its contents.

Parameters:

Name	required	Name of Folder

Example:

```
Delete Folder Name="\Sales Reports"
```

New User

Creates a new User.

Use the **Add Logon** command to add Platform logon(s) to the user.

Parameters:

Name	required	Name of User
Password	optional	Password for User
Role	required	**UserRole**
Email	optional	Email address for User
NTUser	optional	NT Domain\User for User
Preview	optional	Dflt "True". View preview images

Example:

```
New User name="Jane" Password="kju76" Role="Admin"

New User name="Bryan" Password="jhg65" Role="Manager"
email="bb@mycompany.com"
```

Modify User

Modifies an existing user.

Parameters:

Name	required	Name of User
NewName	optional	New name of User
Role	optional	**UserRole**
Email	optional	Email address for User
NTUser	optional	NT Domain\User for User
Preview	optional	Dflt "True". View preview images

Example:

```
Modify User name="Jane" newname="Jayne"

Modify User name="Bryan" email="bb@mycompany.com"
```

Delete User

Deletes an existing user.

Parameters:

Name	required	Name of User

Example:

```
Delete User name="Jane"
```

Add Logon

Add a Platform logon to a User.

Parameters:

User	required	Name of User
Platform	required	Name of ID of Platform
LogonUser	required	UserID for logon
Password	optional	Password for logon
Auth	required	**PlatformAuthType**

Example:

```
Add Logon user="Jane" platform="XIProd" LogonUser="JSmith"
Password="mypassword" Auth="E"
```

Modify Logon

Modify a Platform logon for a User.

Parameters:

User	required	Name of User
Platform	required	Name of ID of Platform
LogonUser	required	UserID for logon
Password	optional	Password for logon
Auth	required	**PlatformAuthType**

Example:

```
Modify Logon user="Jane" platform="XIProd" LogonUser="JSmith"
Password="mynewpassword" Auth="E"
```

Test Logon

Test if a connection can be made to a Platform for a User.

Parameters:

User	required	Name of User
Platform	required	Name of ID of Platform

Example:

```
Test Logon user="Jane" platform="XIProd"
```

Delete Logon

Delete a Platform logon for a User.

Parameters:

User	required	Name of User
Platform	required	Name of ID of Platform

Example:

```
Delete Logon user="Jane" platform="XIProd"
```

Add DBLogon

Add a Database logon to a User. This is typically required for Crystal Documents.

Parameters:

User	required	Name of User
Server	required	Name of Server
Database	required	Name of Database
LogonUser	required	UserID for logon
Password	optional	Password for logon

Example:

```
Add DBLogon user="Jane" server="SQLProd" database="Reports"
LogonUser="JSmith" Password="mypassword"
```

Modify DBLogon

Modify a Database logon for a User.

Parameters:

User	required	Name of User
Server	required	Name of Server
Database	required	Name of Database
LogonUser	required	UserID for logon
Password	optional	Password for logon

Example:

```
Modify DBLogon user="Jane" server="SQLProd" database="Reports"
LogonUser="JSmith" Password="mynewpassword"
```

Delete DBLogon

Delete a Database logon for a User.

Parameters:

User	required	Name of User
Server	required	Name of Server
Database	required	Name of Database

Example:

```
Delete DBLogon user=”Jane” server=”SQLProd” database=”Reports”
```

List UserCache

List cached items for a User.

The Var parameter will create a variable that contains the list of ID's that can be used with the Delete UserCache command (to save having to type them in).

To find cache items linked to a specific object use the option "O:n" where n is the ID of the Object.

Parameters:

User	required	Name of User
Type	optional	**UserCacheContent**
Options(Opt)	optional	"O:id" or "K:prefix"
Var	optional	Name or Variable to assign ID's to

Example:

```
List UserCache user="Jane" type="file"

List UserCache user="Jane" type="file" Opt="O:100" var="docs"
```

Delete UserCache

Delete cached items for a User.

If you wish to delete multiple items first use the above command and use the "Var" option to build a variable that contains the ID's.

Parameters:

User	required	Name of User
Mode	optional	"**selected**" or "all"
ID	required	ID or List to delete

Example:

```
Delete UserCache user="Jane" ID="$docs"
```

New Group

Creates a new Group.

Use the **Add Member** command to add users to the group.

If you wish to change the name of the Group at a later time, use the **RENAME** command.

Parameters:

Name	required	Name of Group
Priority	optional	Highest Schedule Priority (0-99)

Example:

```
New Group name="Sales" Priority="50"
```

Modify Group

Modifies an existing group.

Parameters:

Name	required	Name of Group
NewName	optional	New name of Group
Priority	optional	Highest Schedule Priority (0-99)

Example:

```
Modify Group name="Sales" Priority="60"
```

Delete Group

Delete an existing Group.

Parameters:

Name	required	Name of Group

Example:

```
Delete Group name="Sales"
```

Add Member

Add a User to a Group.

Parameters:

Group	required	Name of Group
User	required	Name of User

Example:

```
Add Member Group="Sales" User="Rick"
```

Delete Member

Remove a User from a Group.

Parameters:

Group	required	Name of Group
User	required	Name of User

Example:

```
Delete Member Group="Sales" User="Rick"
```

ACL

Allow or Deny access to a Folder for one or more Groups.

If just a Folder is given, the ACL will be displayed.

Parameters:

Folder	required	ID or Path for Folder
Group(s)	optional	One of more Group Name/ID (1,2)
Allow	optional	“view” or “create”
Deny	optional	“view” or “create”

Example:

```
ACL Folder=”\Sales” Group=”Sales” Allow=”create”

ACL Folder=”\Common” Groups=”Sales,Marketing” Allow=”view”
```

Cataloging Documents

These commands allow you to add new documents to the InfoBurst repository which can then be used in a 'burst' to refresh and publish the content.

The main purpose of 'cataloging' is to discover information about the document such as its name, ID, location and for a BI document any parameters/prompts/report-tabs that may exist.

Documents can be cataloged from either the filesystem or from a supported platform such as BusinessObjects XIR2.

InfoBurst supports the following documents:

➢ Desktop Intelligence (.rep)	FileSystem and XI Platform
➢ Crystal Reports 11+ (.rpt)	FileSystem and XI Platform
➢ PDF (.pdf)	FileSystem and XI Platform
➢ Microsoft Word (.doc)	FileSystem and XI Platform
➢ Microsoft Excel (.xls)	FileSystem and XI Platform
➢ Microsoft Powerpoint (.ppt)	FileSystem and XI Platform
➢ Crystal Xcelsius or Flash (.swf)	FileSystem and XI Platform
➢ Zip archive (.zip)	FileSystem

To Catalog a BusinessObjects document, you need to ensure that a Platform has been created and a Logon exists for the current user so that a connection can be established.

There are two formats of the Catalog command, one when you wish to catalog from a server platform and the other for local files.

Once the document has been cataloged, you can use the standard **DISPLAY** command to see details about the new Document object.

Catalog (Version to use with an Enterprise Platform)

Catalogs a new document.

The Platform and DocID must be given to identify the document to catalog.

The DocID can be found by using the **Plist Folder="folder" Platform="platform"** command.

To wait for each Catalog operation to be processed in-line, specify **BGMode="false".**

If no folder is given, the current folder will be used.

Parameters:

Platform	required	Name of Platform
Document	required	Document ID or Path\Name
BGMode	optional	"True" to run in background (default)
Folder	optional	Folder to store document

Example:

```
1     Catalog Platform="XIProd" Document="4619" Folder="\Sales
Reports\DivA"

2     Catalog Platform="XIProd" Document="Sales\WklyReport"
```

Example 1 catalogs a document with a known ID and stores the document in a specific folder.

Example 2 catalogs a document using the path\name of the document on the platform. The document will be cataloged into the current folder.

Catalog (Version to use with a local or network File)

Catalogs a new document.

To Catalog a file such as a PDF/Word/Excel all you need is to specify the File.

To Catalog a BusinessObjects Desktop Intelligence report (.rep) you will need to specify a suitable client Platform (such as DI20 or DI30) and have a suitable platform logon setup.

If no folder is given, the current folder will be used.

If no name is given, the current filename will be used (without the extension).

If you wish for InfoBurst to "link" to the file at runtime, set Link="1". This means that InfoBurst will attempt to copy the file from the original location each time it is used. This option will only work when the file path is a suitable UNC style path such as \\ReportServer\Reports\Sales.rep or the file is stored on a shared network folder using a drive letter that exists also on the InfoBurst server.

Parameters:

File	required	File to catalog
Platform	optional	Name of Platform
Folder	optional	Folder to store document
Name	optional	Name of new object
Link	optional	set to "1" to link to file

Example:

```
1     Catalog File="c:\MyDocs\Sales.pdf"

2     Catalog File="c:\MyDocs\Sales.rep" Platform="DeskI"
```

Example 1 catalogs a local PDF document and stores the document in the current folder.

Example 2 catalogs a local Desktop Intelligence report using a defined platform.

ReCatalog

Re-catalogs an existing Document. This is required if the parameters have changed (new parameters added or old parameters removed).

Parameters:

Document	required	Document ID or Path\Name
BGMode	optional	"True" to run in background
File	optional	Local file (if network and not linked)

Example:

```
1     Recatalog Document="\Sales\RegionReport"
```

Wait Document(Doc)

This command will wait for the last Cataloged Document to be processed to allow a script to be written that could perform an action afterwards such as adding the new Document to a Burst etc.

The default delay is "2" seconds.

Parameters:

Delay	optional	# seconds between checking status

Example:

```
Wait Document

Wait Document Delay="1"
```

Modify Document

Modify an existing Document.

It is also possible to change a parameter's value or default value.

If using a name rather than an ID, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Document
NewName	optional	New name for the Object
Description	optional	Description
LinkTo	optional	New network path
Param	optional	Parameter to modify
ParamDefault	optional	New Parameter default value
ParamDescription	optional	New Parameter description
Preview	optional	Dflt "True". Create preview images

Example:

```
Modify Document Name="SalesReport" Description="Latest 2008 Version"

Modify Document Name="SalesReport" Param="Year" ParamDefault="2008"
ParamDescription="Enter a Year between 2006 and 2008"
```

Delete Document

Delete an existing Document.

The object can be located using either the full path\name or a unique ID. If using a name and no path is given then the path will default to the current folder.

Parameters:

Name	optional	Path\Name of Document
ID	optional	Unique ID of Document

Example:

```
Delete Document Name="SalesReport"

Delete Document Name="\Sales\SalesReport"

Delete Document ID="1234"
```

Display LOV

Display any list-of-values for a parameter of a Document.

If using a name rather than an ID, if no path is given then the path will default to the current folder.

Parameters:

Document	required	ID or Path\Name for Document
Parameter	required	Name of Parameter

Example:

```
Display LOV Document="SalesReport" Parameter="Division"
```

Test DocQry

Tests if a given Database Query can satisfy the parameter/filter values for the document.

Parameters:

Document	required	ID or Path\Name of Document
DBConn	required	ID or Path\Name of Connection
DBQry	required	Name of Query

Example:

```
Test DocQry Document="SalesReport" DBConn="MySQLServer" DBQry="Q1"
```

Database Connections

These commands allow you to create/modify/delete database connections and queries.

When working with queries, there is an available shortcut syntax whereby if a command needs both DBConn and Name (as most query commands do) you can include both in the DBConn value separated by a period.

As DBConn is the default parameter for the query commands (display,modify,delete,run) the following two commands are the same:

Display query dbconn=”MyConnection” name=”MyQuery”

Display query “MyConnection**.**MyQuery”

New DBConn / Replace DBConn

Create or Replace a Database Connection. If Replace command used any existing object with the same name in the same Folder will be deleted first.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of Connection
Type	required	**DBConnectionType**
Server	optional	Name of Server
User	optional	User for database logon
Password	optional	Password for database logon
SqlServerCatalog	optional	Catalog name if SqlServer
MSAccessSystemDB	optional	System database for MS Access
Timeout	optional	Max # seconds for Query
Folder	optional	Folder to store connection.

Example:

```
New DBConn Name="PrdSql" Type="SQLServer" Server="prd2" user="ibpp"
password="xx" sqlservercatalog="customermain"
```

Test DBConn

Check if a connection can be made to the Database.

If using a name for DBConn, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Connection

Example:

```
Test DBConn Name="PrdSql"
```

Modify DBConn

Modify an existing Database Connection. The Type cannot be changed but all other settings can be modified.

If using a name for DBConn, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Connection
NewName	optional	New name for the Object
Server	optional	Name of Server
User	optional	User for database logon
Password	optional	Password for database logon
SqlServerCatalog	optional	Catalog name if SqlServer
MSAccessSystemDB	optional	System database for MS Access
Timeout	optional	Max # seconds for Query

Example:

```
Modify DBConn Name="PrdSql" user="ibpp" password="xx"
sqlservercatalog="customermain"
```

Delete DBConn

Delete an existing Database Connection **and all queries**.

If using a name for DBConn, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Connection

Example:

```
Delete DBConn Name="PrdSql"
```

Add Query(Qry)

Add a new Query to an existing Database Connection.

If using a name for DBConn, if no path is given then the path will default to the current folder.

Parameters:

DBConn	required	Name/ID of Connection
Name	required	Name of Query
Usage	optional	**DBQryUsage**
SQL	required	SQL for Query
Timeout	optional	Max # seconds to Execute

Example:

```
Add Query DBConn="PrdSql" name="Q1" Sql="select * from customers"

Add Query "PrdSql" name="Q2" Usage="EPL" Sql="select custid from
customers where region=@region"
```

Display(Disp) Query(Qry)

Display an existing Query and any parameters for a Database Connection.

If using a name for DBConn, if no path is given then the path will default to the current folder.

To display the whole DBConnection, use the standard **Display** command.

Parameters:

DBConn	required	Name/ID of Connection
Name	required	Name of Query

Example:

```
Display Query DBConn="PrdSql" name="Q1"

Disp Qry "PrdSql.Q1"
```

Modify Query(Qry)

Modify an existing Query for a Database Connection.

If using a name for DBConn, if no path is given then the path will default to the current folder.

To set the DefaultColumn, the query must have been previously executed so that column information can be collected.

If the SQL is changed, any column information will be deleted and therefore you should re-execute the query to ensure it is valid and to rebuild any column information.

Parameters:

DBConn	required	Name/ID of Connection
Name	required	Name of Query
NewName	optional	New name for the Object
Usage	optional	**DBQryType**
SQL	required	SQL for Query
DefaultColumn	optional	Column for single value usage
Timeout	optional	Max # seconds to Execute

Example:

```
Modify Query DBConn=”PrdSql” name=”Q1” Sql=”select * from customers
order by Name”

Modify Qry “PrdSql.Q1” Sql=”select * from customers order by name”
```

Execute(Exec) Query(Qry)

Execute an existing Query for a Database Connection. Column information will be generated and therefore you need to call this function before trying to set the Default Column.

If using a name for DBConn, if no path is given then the path will default to the current folder.

If the Query requires parameters, supply them using **param="value"**

Parameters:

DBConn	required	Name/ID of Connection
Name	required	Name of Query
Limit	optional	# Rows to return (0=All)

Example:

```
Exec Query DBConn="PrdSql" name="Q2" State="AZ"

Execute Qry "PrdSql.Q2" State="AZ"

Execute Qry "PrdSql.Q2" State="AZ" Limit="10"
```

Delete Query(Qry)

Delete an existing Query for a Database Connection.

If using a name for DBConn, if no path is given then the path will default to the current folder.

Parameters:

DBConn	required	Name/ID of Connection
Name	required	Name of Query

Example:

```
Delete Query DBConn="PrdSql" name="Q1"

Delete Qry "PrdSql.Q1"
```

Ldap Connections

These commands allow you to create/modify/delete ldap connections.

New LdapConn / Replace LdapConn

Create or Replace an Ldap Connection. If Replace command used any existing object with the same name in the same Folder will be deleted first.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of Connection
Server	required	Name of Server
Port	optional	Default is “389”
User	optional	Domain\User for logon (if required)
Password	optional	Password for logon (if required)
MaxResults	optional	Default is “0” (all)
Timeout	optional	Default is “60” seconds
Folder	optional	Folder to store connection.

Example:

```
New LdapConn Name=”Exchange” Server=”msxch1” user=”infosol\ibpp” password=”xx”
```

Modify LdapConn

Modify an existing Ldap Connection.

If using a name for LdapConn, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Connection
NewName	optional	New name for the Object
Server	optional	
Port	optional	
User	optional	
Password	optional	
MaxResults	optional	
Timeout	optional	

Example:

```
Modify LdapConn Name="PrdSql" user="ibpp" password="xx"
sqlservercatalog="customermain"
```

Delete LdapConn

Delete an existing Ldap Connection. If any LdapList object is using the connection the delete will fail.

If using a name for LdapConn, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Connection

Example:

```
Delete LdapConn Name="PrdSql"
```

Lists

These commands allow you to create/modify/delete simple lists that can be used to supply document parameters or email distribution lists.

To view a List, use the standard display command (e.g. display list “myList”)

To rename or delete a List, use the standard Rename/Delete commands.

New List / Replace List

Create or Replace a List. If Replace command used any existing object with the same name in the same Folder will be deleted first.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of List
Values	required	Delimited list of values
Delim	optional	Delimiter to use (default is “,”)
Folder	optional	Folder to store list.

Example:

```
New List Name=”States” values=”AZ,CA”

New List Name=”Accounts” delim=”/” values=”1001/1002”

Replace List Name=”States” values=”NY,NJ”
```

Modify List

Modify an existing List.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List
NewName	optional	New name for the Object
Add	optional	Delimited list of Items to add
Remove	optional	Delimited list of Items to remove
Delim	optional	Delimiter to use (default is ",")

Example:

```
Modify List Name="States" Add="NY,NJ"

Modify List Name="States" Remove="CA"
```

Delete List

Delete an existing List.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List

Example:

```
Delete List Name="States"
```

File Lists

These commands allow you to create/modify/delete file lists that can be used to supply document parameters or email distribution lists.

To view a FileList, use the standard display command (e.g. display filelist “myList”)

To rename or delete a FileList, use the standard Rename/Delete commands.

New FileList / Replace FileList

Create or Replace a FileList. If Replace command used any existing object with the same name in the same Folder will be deleted first.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of List
File	required	Source File or location on Platform
Platform	optional	ID or Name of Platform
DataColumn	optional	Column name to return (if Excel)
DataCondition	optional	Filter to apply (if Excel)
MaxResults	optional	Max # results to return
ParseEmail	optional	set to “True” for email syntax
Folder	optional	Folder to store list

Example:

```
New FileList Name=”States” file=”\\share1\data\states.txt”

New FileList Name=”AZ_Accounts” file=”g:\data\accounts.xls”
datacolumn=”customer” datacondition=”state=’AZ’”
```

Modify FileList

Modify an existing FileList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List
NewName	optional	New name for the Object
Source	optional	
DataColumn	optional	
DataCondition	optional	
MaxResults	optional	
ParseEmail	optional	

Example:

```
Modify FileList Name="States"
source="\\share1\data\western_states.txt"
```

Delete FileList

Delete an existing FileList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List

Example:

```
Delete FileList Name="States"
```

MultiColumn Lists

These commands allow you to create/modify/delete multi-column lists that can be used to supply document parameters or email distribution lists.

To view a MultiColumnList, use the standard display command (e.g. display multicolumnlist “myList”)

To rename or delete a MulticolumnList, use the standard Rename/Delete commands.

New MultiColumnList(MCList) / Replace MultiColumnList(MCList)

Create or Replace a MultiColumnList. If Replace command used any existing object with the same name in the same Folder will be deleted first.

If no folder is given, the current folder will be used.

To create a list with matching columns to that of an existing Document, specify the Document using the **FromDoc** parameter.

Parameters:

Name	required	Name of List
Columns	optional	List of initial string columns
FromDoc	optional	Name or ID of Document as base
D**elim**	optional	Column delimiter (dflt = “,”)
Folder	optional	Folder to store list

Example:

```
New MultiColumnList Name=”MyList” Columns=”State,City”

New MCList Name=”BurstingList” Delim=”;” Columns=”Division;Region”

New MCList Name=”Customers” FromDoc=”\Docs\Sales\WeeklySales”
```

Modify MultiColumnList(MCList)

Modify an existing MultiColumn List.

Parameters:

Name	required	Name/ID of List
NewName	optional	New name for the Object
DefaultColumn	optional	Column name for single values
DataCondition	optional	Filter condition
SortBy	optional	Column(s) to sort “a,b”

Example:

```
Modify MultiColumnList Name=”MyList” DefaultColumn=”State”
```

Delete MultiColumnList(MCList)

Delete an existing MultiColumnList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List

Example:

```
Delete MultiColumnList Name=”States”
```

Add Column

Add a column to an existing MultiColumn List.

Parameters:

MCList	required	Name/ID of MultiColumn List
Name	required	Name of the new Column
Type	optional	**ListColumnType**
SetExisting	optional	Value for existing rows

Example:

```
Add Column MCList="MyList" Name="Store"
```

Add Columns

Add columns to an existing MultiColumn List that match all prompts/filters from a Document.

Parameters:

MCList	required	Name/ID of MultiColumn List
Document	required	Name/ID of the Document

Example:

```
Add Columns MCList="MyList" Document="\Sales\Reports\QTR"
```

Add Row

Add a row to an existing MultiColumn List. Enter the values as a delimited string in column number order.

If no Position is given, the Row will be added at the end.

Parameters:

MCList	required	Name/ID of MultiColumn List
Values	required	List of values to add

Example:

```
Add Row MCList="MyList" values="Arizona,Phoenix"
```

Delete Column

Delete a Column from a MultiColumn List.

Parameters:

MCList	required	Name/ID of MultiColumn List
Name	required	Column name

Example:

```
Delete Column MCList="MyList" name="City"
```

Delete Row

Delete a Row from a MultiColumn List.

Parameters:

MCList	required	Name/ID of MultiColumn List
Number	required	Row #

Example:

```
Delete Row MCList="MyList" Number="10"
```

Ldap Lists

These commands allow you to create/modify/delete Ldap lists that can be used to supply document parameters or email distribution lists.

To view an Ldap ist, use the standard display command (e.g. display ldaplist “myList”)

To rename or delete an LdapList, use the standard Rename/Delete commands.

New LdapList / Replace LdapList

Create or Replace an LdapList. If Replace command used any existing object with the same name in the same Folder will be deleted first.

The default SearchScope is “OneLevel”.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of List
Connection	required	Path\Name for an LdapConnection
DN	required	DN root for search
SearchFilter	required	Filter for search
SearchScope	optional	**`LdapSearchScope`**
DataAttribute	optional	Default is “mail”
DataCondition	optional	Filter to apply
MaxResults	optional	Max # results to return
Folder	optional	Folder to store list

Example:

```
New LdapList Name=”Infosol” Connection=”\Connections\Exchange”
DN=”cn=Users,DC=INFOSOL,DC=COM” SearchFilter=”sn=W*”
```

Modify LdapList

Modify an existing LdapList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List
NewName	optional	New name for the Object
Connection	optional	
DN	optional	
SearchFilter	required	
SearchScope	optional	**LdapSearchScope**
DataAttribute	optional	
DataCondition	optional	
MaxResults	optional	

Example:

```
Modify LdapList Name="Infosol" searchfilter="sn=D*"
```

Delete LdapList

Delete an existing LdapList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List

Example:

```
Delete LdapList Name="Infosol"
```

XI Lists

These commands allow you to create/modify/delete XI lists that can be used to retrieve XI user account names or email addresses for use with distribution lists.

To view an XI list, use the standard display command (e.g. display xilist “myList”)

To rename or delete an XIList, use the standard Rename/Delete commands.

New XIList / Replace XIList

Create or Replace an XIList. If Replace command used any existing object with the same name in the same Folder will be deleted first.

The default DataAttribute is “email”.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of List
Platform	required	Path\Name for a Platform
Groups	required	List of Groups (a,b)
DataAttribute	optional	“email”, “name”, “fullname”
MaxResults	optional	Max # results to return
Folder	optional	Folder to store list

Example:

```
New XIList Name=”XIEmail” Platform=”XIR2” Groups=”IBDev,IBSup”
DataAttribute=”email”

New XIList Name=”XIName” Platform=”XIR2” Groups=”IBDev,IBSup”
DataAttribute=”name”
```

Modify XIList

Modify an existing XIList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List
NewName	optional	New name for the Object
Platform	required	Path\Name for a Platform
Groups	required	List of Groups (a,b)
DataAttribute	optional	"email", "name", "fullname"
MaxResults	optional	Max # results to return

Example:

```
Modify XIList Name="XIEmail" groups="Sales,Support"
```

Delete XIList

Delete an existing XIList.

If using a name for a List, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of List

Example:

```
Delete XIList Name="XIEmail"
```

Templates

These commands allow you to create/modify and delete Delivery Templates.

To view a Template, use the standard display command (e.g. display burst “myBurst”)

To rename or delete a Template, use the standard Rename/Delete commands.

New Template / ReplaceTemplate

Create or Replace a Template. If Replace command used any existing object with the same name in the same Folder will be deleted first.

A Template is a pre-defined list of settings that can be used as a Burst Delivery and therefore every option that can be used with the “**Add Delivery**” command can be used here with the exception of anything that requires specific Document knowledge (such as Report Tabs).

Examples:

```
New Template Name=”Network PDF To=”\\prod\reports\2012”

New Template Name=”Email HTML” Format=”html” Dest=”Email”
To=usr1@infosol.com,usr2@infosol.com
```

Bursts

These commands allow you to create/modify/delete/execute Bursts and manage the documents and distributions etc.

To view a Burst, use the standard display command (e.g. display burst “myBurst”)

To rename or delete a Burst, use the standard Rename/Delete commands.

New Burst / Replace Burst

Create or Replace a Burst. If Replace command used any existing object with the same name in the same Folder will be deleted first.

GenMode defaults to Sequential, **DelMode** defaults to Immediate.

GroupEmail is a Boolean and defaults to “True”.

Email can be grouped by recipient at Document level (DelMode=Immediate) or Burst level (DelMode=AtEnd).

EmailIfAbort is a Boolean and defaults to “False”.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of Burst
Description	optional	Description
GenMode	optional	**`RTSGenerationMode`**
DelMode	optional	**`RTSDeliveryMode`**
GroupEmail	optional	Consolidate Email
EmailIfAbort	optional	Email the person who runs the burst?
EmailSubject	optional	Custom Subject
EmailFrom	optional	Custom from address
EmailFooter	optional	Custom email footer
EmailTOC	optional	Generate Table of Contents?

EmailMessage	optional	Message at top/bottom
EmailMessageOnTop	optional	“true” for Top, “false” for bottom
OnStart	optional	Initial list of Start Actions “ac1,ac2”
OnComplete	optional	Initial list of Complete Actions “ac1,ac2”
OnAbort	optional	Initial list of Abort Actions “ac1,ac2”
Delim	optional	Delimiter for Actions “,”
Lookup	optional	ID/Name/Path for FileList Object
DateLookup	optional	ID/Name/Path for FileList Object
RTSLog	optional	Enable detailed RTS Log
SortAlpha	optional	“true” to sort Docs in Burst A-Z
Folder	optional	Folder to store list

Example:

```
New Burst Name=”SalesBurst”

New Burst Name=”SalesBurst” GenMode=”PAR”
```

Modify Burst

Modify an existing Burst.

If using a name for a Burst, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Burst
NewName	optional	New name for the Object
Description	optional	Description
GenMode	optional	**`RTSGenerationMode`**
DelMode	optional	**`RTSDeliveryMode`**
GroupEmail	optional	Consolidate Email
EmailIfAbort	optional	Email the person who runs the burst?
EmailSubject	optional	Custom Subject
EmailFrom	optional	Custom from address
EmailFooter	optional	Custom email footer
EmailTOC	optional	Generate Table of Contents?
EmailMessage	optional	Message at top/bottom
EmailMessageOnTop	optional	“true” for Top, “false” for bottom
OnStart	optional	List of Start Actions “ac1,ac2”
OnComplete	optional	Llist of Complete Actions “ac1,ac2”
OnAbort	optional	Llist of Abort Actions “ac1,ac2”
Delim	optional	Delimiter for Actions “,”
RTSLog	optional	Enable detailed RTS Log
SortAlpha	optional	“true” to sort Docs in Burst A-Z

Example:

```
Modify Burst Name="SalesBurst" DelMode="Burst" EmailIfAbort="true"
```

Delete Burst

Delete an existing Burst.

If using a name for a Burst, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Burst

Example:

```
Delete Burst Name="SalesBurst"
```

Add Document

Adds a Document to a Burst.

Normally a Document is first "Cataloged" whereby any prompts/filters etc will be discovered and the resulting Document Object can be used in one or more Bursts.

It is possible to add a "Live" Document to a Burst whereby the Document has not been previously Cataloged. To do this, specify a PlatformID that corresponds to an XIR2 or XI 3.0 platform and a DocID that corresponds to the ID of the Document on the Platform.

Defaults will be set to "True" if not specified. This will copy the last known parameter values (from when the Document was last Cataloged).

If using a name for a Burst, if no path is given then the path will default to the current folder.

Parameters:

Burst	required	Name/ID of Burst
PlatformID	optional	Name/ID of Platform (for Live)
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document
Defaults	optional	"True" to copy Catalog defaults

Example:

```
Add Document Burst="SalesBurst" DocID="Sales per Region"

Add Document Burst="SalesBurst" DocID="Sales per Region" Instance="SPR West" defaults="false"

Add Document Burst="SalesBurst" PlatformID="XIR2" DocID="4336"
```

Remove Document

Removes a Document from a Burst. Any Parameters and Deliveries will also be removed.

If using a name for a Burst, if no path is given then the path will default to the current folder.

Parameters:

Burst	required	Name/ID of Burst
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document

Example:

```
Remove Document Burst="SalesBurst" DocID="Sales per Region"
```

Set Document

Sets a Source for a Document within a Burst. The Source can be either a MultiColumn List or a Database Query.

For a SourceType of "Qry" (DataBase Query) specify "DBConnection**.**DBQry"

Parameters:

Burst	required	Name/ID of Burst
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document
SourceType	optional	**`BurstParameterSource`**
Source	optional	Name/ID of Source List/Qry
SourceParams	optional	Parameters for Source
SourceParamDelim	optional	Delimiter if multi-values

Examples:

```
Set Document Burst="B1" DocID="SPR" SourceType="Qry"
Source="SQL1.StateInfo"
```

Set Parameter

Sets a Parameter value or Source for a Document within a Burst.

By default, the SourceType will be “Inline” which means you can use the **Value** to supply one or more values.

The default MVDelim is “;”.

The default for MultiPass is “True”.

To use an external source such as a List/FileList/MultiColumnList or Database Query, set the SourceType to the desired value and then specify the Name/ID of the Source.

For a SourceType of “Qry” (DataBase Query) specify “DBConnection**.**DBQry”

For simple Source assignments you can specify a Source in the Value such as “list:MyList” or “mcl:StateInfo(column)” instead of using the SourceXX options.

Parameters:

Burst	required	Name/ID of Burst
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document
Name	required	Name of Parameter
Value	optional	Value(s) or Source:Params
MVDelim	optional	Delimiter if multi-values
MultiPass	optional	“True” to split each value
SourceType	optional	**`BurstParameterSource`**
Source	optional	Name/ID of Source List/Qry
SourceParams	optional	Parameters for Source
SourceParamDelim	optional	Delimiter if multi-values

Examples:

```
Set Parameter Burst="B1" DocID="SPR" Name="State" Value="Arizona"

Set Parameter Burst="B1" DocID="SPR" Name="State"
Value="Arizona;California" MultiPass="True"

Set Parameter Burst="B1" DocID="SPR" Name="State" SourceType="List"
Source="WesternStates"

Set Parameter Burst="B1" DocID="SPR" Name="State" SourceType="Qry"
Source="SQL1.WesternStates"
```

Add Filter

Adds a Filter to a Document in a Burst.

A Filter allows for single-pass bursting and often leads to increased runtime performance.

If using a name for a Burst, if no path is given then the path will default to the current folder.

The Name must be a valid Report Element that was found when the Document was Cataloged.

The Source will be set to match the Report Element ("ReportVariable") and therefore at run-time all values will be extracted from the report and a separate report will be generated for each unique value.

If you wish to supply your own values for the filter, use the **Set Filter** command.

Parameters:

Burst	required	Name/ID of Burst
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document
Name	required	Name of Report Element

Example:

```
Add Filter Burst="SalesBurst" DocID="Sales per Region" Name="City"
```

Set Filter

Sets a Filter value or Source for a Document within a Burst.

By default, the SourceType will be “ReportVariable” and therefore you can change the source type to “inline” or any other source that you wish to use.

The default MVDelim is “;”.

To use an external source such as a List/FileList/MultiColumnList or Database Query, set the SourceType to the desired value and then specify the Name/ID of the Source.

For a SourceType of “Qry” (DataBase Query) specify “DBConnection**.**DBQry”

For simple Source assignments you can specify a Source in the Value such as “list:MyList” or “mcl:StateInfo(column)” instead of using the SourceXX options.

Parameters:

Burst	required	Name/ID of Burst
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document
Name	required	Name of Filter
Value	optional	Value(s) or Source:Params
MVDelim	optional	Delimiter if multi-values
SourceType	optional	**`BurstParameterSource`**
Source	optional	Name/ID of Source List/Qry
SourceParams	optional	Parameters for Source
SourceParamDelim	optional	Delimiter if multi-values
SourceColumn	optional	Column if not the default
Combine	optional	“True” to use all values as one

Examples:

```
Set Filter Burst="B1" DocID="SPR" Name="City" SourceType="Inline"
Value="Phoenix;Tucson;Scottsdale"

Set Filter Burst="B1" DocID="SPR" Name="City" SourceType="List"
Source="CityList"
```

Add Delivery

Adds a Delivery to a Document in a Burst.

A Delivery is a combination of a Format (Such as PDF or Excel) and a Destination (Such as Network or Email) and any other specific options that are required.

Build 118 and beyond support Delivery Templates and so it is possible to create one or more of these Templates and then use them instead of having to create specific Deliveries.

To use a Template, specify the Name or the ID of the Template using the **Template** keyword.

The default Format is PDF and the default Destination is Network.

If no Name is given, a name will be assigned based on the destination and format.

All Network shares must be registered first. See "Register Path".

Parameters:

Name	optional	Name of Delivery
Burst	required	Name/ID of Burst
DocID	required	Name/ID of Document
Instance	optional	Instance Name for Document
Format	optional	**`BurstFormat`**
Destination(Dest)	optional	**`BurstDestination`**
AuthUser	optional	User or Domain\User for authentication
AuthPassword	optional	Password for authentication
Message	optional	Message for Email
ReportTabs(Tabs)	optional	List of Tabs to deliver (blank=**All**)
TabDelim	optional	Delimiter if multiple tabs (",")
DeliverTo(To)	optional	Path or Recipients to deliver to
CopyTo(CC)	optional	Recipients to copy to
BlindCopyTo(BCC)	optional	Recipients to blind copy to

Delim	optional	Delimiter if multiple To values (“**,**”)
FileName	optional	Name of generated File
Zip	optional	Name of generated Zip Archive
ZipPassword	optional	Password to open Zip
ZipEmail	optional	Zip attachments? (“true” or “**false**”)
Combine	optional	Combine Tabs? (“true” or “**false**”)
Replace	optional	Replace if exists? (“true” or “**false**”)
Categories	optional	List of XI Categories
LinkDeliverTo(LTo)	optional	Recipients to deliver content links to
LinkCopyTo(LCC)	optional	Recipients to copy content links to
LinkBlindCopyTo(LBCC)	optional	Recipients to blind copy content links to
FTPServer	optional	Name of FTP Server
FTPPort	optional	Port # if not default
PdfXOffset(PdfX,X)	optional	Horiz offset for Pdf Print
PdfYOffset(PdfY,Y)	optional	Vertical offset for Pdf Print
PdfFirstPage(First)	optional	Page # to start printing
PdfLastPage(Last)	optional	Page # to stop printing
PdfOptions(Options)	optional	Pdf Print options (“#NOSCALE”)
CustomCommand(Cmd)	optional	Path to script or command
CustomParams(Params)	optional	Params for command (“%F”)
XIPlatform	optional	XI Platform ID/Name if Pdf or Excel
TextDelim	optional	Delimiter for Text (if Fmt=TextDelim)
TextExtension	optional	File Extension for Text format
TextHeader	optional	Delim Header ? (“**true**” or “false”)
ScaleToFit(Scale)	optional	Scale SWF (Email) ? (“true” or “**false**”)

Page(Pages)	optional	For PNG, pages to display “1”, “1,2-3”
Width(w)	optional	for PNG, width of image
Height(h)	optional	For PNG, height of image
Condition(Cond)	optional	Condition (e.g. State=’Arizona’)
XLAutoFilter(xlaf)	optional	List of worksheets (e.g. “S1,S2”)
XLGroup(xlgrp)	optional	List of worksheets
XLGroupColumn(xlgrpcol)	optional	Column to Group (dflt=”A”)
XLGroupHeader(xlgrphdr)	optional	Group header (“true” or “**false**”)
XDCToEmbed(xdc)	optional	For SWF/Native, XmlDataCache to use
RoamBiServer	optional	RoamBi ES Server
RoamBiPortal	optional	RoamBI Portal
RoamBiTemplate	optional	RoamBi template RBI on Portal
SSLStore	optional	Name of cert store for FTPS
SSLStorePassword	optional	Password for store
SSLCert	optional	Name of cert
IgnoreErrors	optional	Ignore SSL connection errors for FTPS

Recipients can be entered as follows:

- user@company.com
- [%macro]
- **user:**name
- **group:**name
- **list:**path\name or **list:**name
- **filelist:**path\name or **filelist:**name
- **ldaplist:**path\name or **ldaplist:**name

- **xilist:**path\name or **xilist:**name
- **mclist:**path\name or **mclist:**name
- **qry:**path\connection.qry or **qry:**connection.qry

A Column-name can be supplied (to over-ride any default) between ( ) and any conditions or query parameters can be supplied between { } for the following types:

- Filelist
- Ldaplist
- Mclist
- Qry

Examples:

```
Add Delivery Burst="SalesBurst" DocID="Sales per Region"
To="\\prod\reports\2008" FileName="Sales for [=City]"

Add Delivery Burst="SalesBurst" DocID="Sales per Region" Format="html"
Dest="Email" To=usr1@infosol.com,usr2@infosol.com

Add Delivery Burst="SalesBurst" DocID="Sales per Region" Format="html"
Dest="Email" To="user:john,group:sales"

Add Delivery Burst="SalesBurst" DocID="Sales per Region" Format="html"
Dest="Email" To="list:Sales\SalesManagers"

Add Delivery Burst="SalesBurst" DocID="Sales per Region" Format="html"
Dest="Email" To="filelist:XLManagers(RegionEmail)"

Add Delivery Burst="SalesBurst" DocID="Sales per Region" Format="html"
Dest="Email" To="mclist:SalesManagers(Email){State='[=state]'}"

Add Delivery Burst="MyBurst" DocID="Weekly Sales" Format="Xls"
To="c:\[$Burst]" FileName="Sales Summary" xlaf="Sheet1,Sheet2"
xlgrp="Sheet3"

Add Delivery Burst="MyBurst" DocID="Weekly Sales" Format="Pdf"
To="c:\[$Burst]" FileName="Sales Summary [=Region]" Condition="States
in('Arizona','California')"
```

Add Recipient

Adds one or more new recipients to an existing Delivery for a Document in a Burst.

Parameters:

Burst	required	Name/ID of Burst
Document	required	Name/ID of Document within Burst
Instance	optional	Instance Name for Document
Delivery	required	Name/ID of Delivery for Document
DeliverTo(To)	optional	Path or Recipients to deliver to
CopyTo(CC)	optional	Recipients to copy to
BlindCopyTo(BCC)	optional	Recipients to blind copy to
Delim	optional	Delimiter if multiple To values (“**,**”)
LinkDeliverTo(LTo)	optional	Recipients to deliver content links to
LinkCopyTo(LCC)	optional	Recipients to copy content links to
LinkBlindCopyTo(LBCC)	optional	Recipients to blind copy content links to

Remove Recipient

Removes one or more recipients from an existing Delivery for a Document in a Burst.

Parameters:

Burst	required	Name/ID of Burst
Document	required	Name/ID of Document within Burst
Instance	optional	Instance Name for Document
Delivery	required	Name/ID of Delivery for Document
DeliverTo(To)	optional	Path or Recipients to deliver to
CopyTo(CC)	optional	Recipients to copy to
BlindCopyTo(BCC)	optional	Recipients to blind copy to
Delim	optional	Delimiter if multiple To values (“**,**”)
LinkDeliverTo(LTo)	optional	Recipients to deliver content links to
LinkCopyTo(LCC)	optional	Recipients to copy content links to
LinkBlindCopyTo(LBCC)	optional	Recipients to blind copy content links to

QuickBurst(Burst)

This command creates a burst in a single step and is useful to quickly create a burst based on a single document that uses a common delivery type such as PDF to the Network or HTML via Email.

The default Name is “MyQuickBurst”. Use Name=”x” to give a specific name to the Burst.

Use Replace=”true” to delete any existing Burst of the same name first.

The default MVDelim is “;”.

The default for MultiPass is “True”.

The default Format is “PDF”.

To set a Parameter, specify Param=”Value”.

The following values can be used to set a Source for a Parameter:

- List:Object
- FileList:Object
- MCList:Object
- Qry:DBConn.Qry

To create a Filter, use Filter=”Variable” to create a Filter that will be based on all values in the report or to specify a list of values manually use Filter=”Variable{value;value}”.

To use email delivery, give one or more email addresses in the “deliverto” delimited with “;”.

Parameters:

Name	optional	Name of Burst to create
Replace(rpl)	optional	“true” to replace burst if exists
Document	required	Name/ID of Document
Description	optional	Description of Burst
MVDelim	optional	Delimiter if multi-values
MultiPass	optional	“True” to split each value
Filter	optional	Report Element for dynamic Filter
Format	optional	“pdf”, ”xls”, ”html”, “text”, “csv”
DeliverTo(deliver,to)	optional	Email recipients
Delim	optional	Delimiter if multiple To (“,”)
Path	optional	Network path to deliver to
FileName	optional	FileName to generate/attachment
Message	optional	Email message
Folder	optional	Network Folder

Examples:

```
Burst Document=”SPR” Name=”MyBurst” State=”Arizona”
DeliverTo=”\\prod\reports”

Burst Document=”SPR” State=”Arizona;California” FileName=”Sales for
[=State]” Deliver=”\\prod\reports”

Burst Document=”SPR” State=”List:WesternStates” FileName=”Sales for
[=State]” to=”\\prod\reports”

Burst Document=”SPR” State=”Qry:SQL1.WesternStates” FileName=”Sales
for [=State]” Deliver=\\prod\reports

Burst Document=”SPR” State=”Arizona” Filter=”City” FileName=”Sales for
[=State] [=City]” Deliver=\\prod\reports
```

```
Burst Document="SPR" State="Arizona" Filter="City" format="html"
To=john@infoburst.com

Burst Document="SPR" State="Arizona" Filter="City" format="html"
To="user:john,group:sales"
```


Set Enabled/Disabled/DisabledNext

This command allows you to turn on/off processing for either a Burst as a whole, a specific Document in a Burst or a specific Delivery for a Document in a Burst.

Parameters:

Burst	required	Name/ID of Burst
DocID	optional	Name/ID of Document
Instance	optional	Instance Name for Document
DeliveryID	optional	ID of Delivery

Examples:

```
Set Disabled Burst="SalesBurst" DocID="Sales per Region"

Set Disabled Burst="SalesBurst"

Set Disabled Burst="SalesBurst" DocID="Sales per Region"
DeliveryID="1456"

Set Enabled Burst="SalesBurst" DocID="Sales per Region"
```

Set ProcMode

This command allows you to change the default Processing Mode for a Document in a Burst.

Parameters:

Burst	required	Name/ID of Burst
DocID	optional	Name/ID of Document
Instance	optional	Instance Name for Document
Mode	required	**DocumentProcessingMode**

Example:

```
Set ProcMode Burst="SalesBurst" DocID="Sales per Region"
mode="ServerOnDemand"
```

Execute(exec) Burst

This command submits a request to start a Burst. Based on the loading on the InfoBurst system the Burst may be queued.

The command will display the runtime ID that can be used to track the progress and/or display detailed actions. The variable "$RTSID" will be set to the runtime ID.

The "To" option will invoke a WAIT BURST and then an EMAIL RTLOG to allow you to run a burst, wait for it to finish and send the runtime log in one command.

Parameters:

Name	required	Name/ID of Burst
To	optional	Email address to send RTS Log
RunAs(user)	optional	InfoBurst user account to use
Password(pwd)	optional	InfoBurst user password

Example:

```
Exec Burst Name="SalesBurst"

Exec Burst 1234 to="jwilcox@infoburst.com"
```

Wait Burst

This command will wait for the last executed Burst to finish to allow a script to be written that could perform an action afterwards such as emailing the runtime log etc.

The default delay is "5" seconds.

Parameters:

Delay	optional	# seconds between checking status

Example:

```
Wait Burst

Wait Burst Delay="2"
```

Export(exp) RTLog(log)

This command will export the current runtime log(s) for the given runtime ID into a .htm file for easier viewing. If you do not know the ID you can specify a Burst name to use the latest runtime ID.

If no ID or Burst is given, the $RTSID will be used (from the last Exec Burst).

By default the order is "time" which will display all actions in the order they happened. If a Burst is processed in Parallel there could be a mixture of actions for different Documents at the same time.

If you prefer to view by Document, specify **Order="doc"** and the output file will be structured based on each Document.

Parameters:

ID	optional	Runtime ID
Burst	optional	Path\Name of Burst
File	required	Path\Name of file to generate
Order	optional	"time" or "doc"

Examples:

```
Export RTLog ID="10" File="c:\temp\burstlog.htm"

Export Log ID="10" File="c:\temp\burstlog.htm" order="doc"

Export Log Burst="MyBurst" File="c:\temp\myburst.log"
```

Email RTLog(log)

This command will email the current runtime log(s) for the given runtime ID to the given recipient. If you do not know the ID you can specify a Burst name to use the latest runtime ID.

If no ID or Burst is given, the $RTSID will be used (from the last Exec Burst).

By default the order is "time" which will display all actions in the order they happened. If a Burst is processed in Parallel there could be a mixture of actions for different Documents at the same time.

If you prefer to view by Document, specify **Order="doc"** and the email will be structured based on each Document.

Parameters:

ID	optional	Runtime ID
Burst	optional	Path\Name of Burst
To	required	Email address
Order	optional	"time" or "doc"
Subject	optional	If blank a default will be used

Examples:

```
Email RTLog ID="10" To="jwilcox@infoburst.com" subject="Burst:Sales"

Email RTLog ID="10" To="sdingley@infoburst.com" order="doc"

Email Log Burst="MyBurst" To="sdingley@infoburst.com"
```

Set BurstAlertMode

This command will set the Burst Alert Mode.

There are three possible modes:

- None No alert emails will be sent
- All An alert email will be sent for EVERY burst abort
- Selected An alert email will be sent for SELECTED burst aborts

To see the current mode, use the “list config” command and look at the value for “BurstAlertMode”.

It is also possible to set the alert mode using set config BurstAlertMode=”x” where x is either 0, 1 or 2 (0=None, 1=All, 2=Selected).

Parameters:

Mode	required	**BurstAlertMode**

Add Alert

This command will add a Burst to the list of **selected** Bursts to monitor in case of an abort.

Parameters:

Burst	required	ID or Path\Name of Burst

Remove Alert

This command will remove a Burst from the list of **selected** Bursts to monitor in case of an abort.

Parameters:

Burst	required	ID or Path\Name of Burst

XDC

These commands allow you to create/modify/delete/execute XDC Objects.

An XDC Object is an XmlDataCache that is used to supply cached data to an Xcelsius dashboard.

To view an XDC, use the standard display command (e.g. display xdc “myBurst”)

To rename or delete a Burst, use the standard Rename/Delete commands.

New XDC / Replace XDC

Create or Replace an XDC. If Replace command used any existing object with the same name in the same Folder will be deleted first.

GenMode defaults to Sequential

EmailIfAbort is a Boolean and defaults to “False”.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of XDC
Description	optional	Description
GenMode	optional	**`RTSGenerationMode`**
EmailIfAbort	optional	Email the person who runs the xdc?
OnStart	optional	Initial list of Start Actions “ac1,ac2”
OnComplete	optional	Initial list of Complete Actions “ac1,ac2”
OnAbort	optional	Initial list of Abort Actions “ac1,ac2”
Delim	optional	Delimiter for Actions “,”
PublishTo(Publish)	optional	Network location to publish Cache
RTSLog	optional	Enable detailed RTS Log
CacheQuery	optional	Enable use of Cache Queries
Folder	optional	Folder to store list

Example:

```
New XDC Name="SalesData"

New XDC Name="SalesData" GenMode="PAR"
```

Modify XDC

Modify an existing XDC.

If using a name for an XDC, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of XDC
NewName	optional	New name for the Object
Description	optional	Description
GenMode	optional	**RTSGenerationMode**
EmailIfAbort	optional	Email the person who runs the xdc?
OnStart	optional	List of Start Actions "ac1,ac2"
OnComplete	optional	List of Complete Actions "ac1,ac2"
OnAbort	optional	List of Abort Actions "ac1,ac2"
Delim	optional	Delimiter for Actions ","
PublishTo(Publish)	optional	Network location to publish Cache
CacheQuery	optional	Enable use of Cache Queries

Example:

```
Modify XDC Name="SalesData" EmailIfAbort="true"
```

Delete XDC

Delete an existing XDC.

If using a name for a XDC, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of XDC

Example:

```
Delete XDC Name="SalesData"
```

Add XDS

Adds a DataSource to the XDC.

Each XDS added to an XDC can pull data from either a Webi Report, Database Query or Excel Workbook.

If you wish to pull multiple parts from the same Webi report simply create multiple XDS objects. When the XDC is refreshed the sources are optimized to ensure that the Webi document is only refreshed once even if multiple sources refer to the same Document.

For a Query you can crosstab the resulting rowset by using the CrossTab commands. This will create a default un-named Crosstab. It is possible to add more Crosstabs using the "Add Crosstab" command.

If using a name for the XDC, if no path is given then the path will default to the current folder.

Parameters:

XDC	required	Name/ID of XDC
Name	required	Name of XDS
Type	required	**XDCType**
Object(doc,dbconn)	optional	ID or Path of Source Object
Item(report,query,url)	required	Report / Query / URL
Part	optional	Part # for WebI Report Tab
AllowRealTime	optional	True/False
MaxRows	optional	#Rows for Xml Cache (2048)

Parameters (Crosstab):

CrossTabX	optional	Column for Crosstab X
CrossTabY	optional	Column for Crosstab Y
CrossTabS	optional	Column for Crosstab Summary
ColumnTotals	optional	**XDSColumnTotals**
RowTotals	optional	**XDSRowTotals**
ColType	optional	**XDSTotalType**
RowType	optional	**XDSTotalType**
NumberFormat	optional	**XDSNumberFormat**
SuppressZero	optional	True/False
AbreviateStates	optional	True/False
AuthUser	optional	User for URL access
AuthPassword	optional	Password for AuthUser

Example:

```
Add XDS Name="Customers" Type="Webi" Doc="XIR2\Sales per Region"
Report="Customer" Part="1"

Add XDS Name="Orders" Type="Query" DBConn="SQL" Query="Q1"
```

Add XCQ

Adds a Cache Query to the XDC.

A Cache Query is a SQL statement that can be run against an existing XDS to produce a limited # rows for use in the dashboard.

To assist in writing the Query, the available column names for the XDS will be listed using the display command (e.g. disp 123 (where 123 is the ID for an XDC)).

The Query should select FROM the table that corresponds to the XDS. For example if your Cache contains an XDS "AllSales" you can write a Query "select * from AllSales .. "

To optimize the performance of the Query you can optionally specify a column to index on and in large caches this could result in much better response. For example if the XDS contains a column "SalesPrice" and your query was "select * from AllSales where SalesPrice>=@Price" you could nominate "SalesPrice" as the index column.

If using a name for the XDC, if no path is given then the path will default to the current folder.

Parameters:

XDC	required	Name/ID of XDC
Name	required	Name of Query
SQL	required	SQL for Query
Index	optional	Column to Index
Columns	optional	Dflt "True". Return Columns
MaxRows	optional	#Rows to return (2048)

Example:

```
Add XCQ Name="Customers" SQL="select * from AllSales where
SalesPrice>=@Price order by SalesPrice"
```

Add Crosstab

Adds a new Crosstab to an XDS or Query.

To add a Crosstab to an XDS, use the XDS keyword to identify the ID/Name.

To add a Crosstab to a Cache Query, use the Query keyword to identify the ID/Name.

Parameters:

XDC	required	Name/ID of XDC
XDS	optional	Name/ID of XDS
Query(XDQ)	optional	Name/ID of Query
Name	required	Name of new Crosstab
CrossTabX(x)	optional	Column for Crosstab X
CrossTabY(y)	optional	Column for Crosstab Y
CrossTabS(s)	optional	Column for Crosstab Summary
ColumnTotals	optional	**XDSColumnTotals**
RowTotals	optional	**XDSRowTotals**
ColType	optional	**XDSTotalType**
RowType	optional	**XDSTotalType**
NumberFormat	optional	**XDSNumberFormat**
SuppressZero	optional	True/False
AbreviateStates	optional	True/False

Modify Crosstab

Change an existing Crosstab for an XDS or Query.

To modify a Crosstab for an XDS, use the XDS keyword to identify the ID/Name.

To modify a Crosstab for a Cache Query, use the Query keyword to identify the ID/Name.

Parameters:

XDC	required	Name/ID of XDC
XDS	optional	Name/ID of XDS
Query(XDQ)	optional	Name/ID of Query
Name	required	Name of Crosstab to Modify
NewName	optional	New Name of Crosstab
CrossTabX(x)	optional	Column for Crosstab X
CrossTabY(y)	optional	Column for Crosstab Y
CrossTabS(s)	optional	Column for Crosstab Summary
ColumnTotals	optional	**XDSColumnTotals**
RowTotals	optional	**XDSRowTotals**
ColType	optional	**XDSTotalType**
RowType	optional	**XDSTotalType**
NumberFormat	optional	**XDSNumberFormat**
SuppressZero	optional	True/False
AbreviateStates	optional	True/False

Remove Crosstab

Remove an existing Crosstab for an XDS or Query.

To remove a Crosstab for an XDS, use the XDS keyword to identify the ID/Name.

To remove a Crosstab for a Cache Query, use the Query keyword to identify the ID/Name.

Parameters:

XDC	required	Name/ID of XDC
XDS	optional	Name/ID of XDS
Query(XDQ)	optional	Name/ID of Query
Name	required	Name of Crosstab to Remove

Delete XDS

Removes an XDS from an XDC.

The Cache will not be updated until a refresh is performed (see Exec XDS).

If using a name for the XDC, if no path is given then the path will default to the current folder.

Parameters:

XDC	required	Name/ID of XDC
Name	required	Name of XDS to Delete

Example:

```
Remove XDS XDC="SalesData" Name="Orders"
```

Delete XCQ

Removes a Cache Query from an XDC.

If using a name for the XDC, if no path is given then the path will default to the current folder.

Parameters:

XDC	required	Name/ID of XDC
Name	required	Name of Query to Delete

Example:

```
Remove XCQ XDC="SalesData" Name="LatestOrders"
```

Set ParamSource

Sets a Source for all Parameters within an XDC. The Source can be either a MultiColumn List or a Database Query.

For a SourceType of “Qry” (DataBase Query) specify “DBConnection**.**DBQry”

Parameters:

XDC	required	Name/ID of XDC
SourceType	optional	**`BurstParameterSource`**
Source	optional	Name/ID of Source List/Qry
SourceParams	optional	Parameters for Source
SourceParamDelim	optional	Delimiter if multi-values

Examples:

```
Set ParamSource XDC=”MyCache” SourceType=”Qry”
Source=”SQL1.MyCustomers”
```

Set Parameter

Sets a Parameter value to all matching within an XDC.

The default MVDelim is ";".

The default for MultiPass is "True".

Parameters:

XDC	required	Name/ID of XDC
Name	required	Name of Parameter
Value	optional	Value(s) or Source:Params
MVDelim	optional	Delimiter if multi-values
MultiPass	optional	"True" to split each value

Example:

```
Set Parameter XDC="MyCache" Name="State" Value="Arizona"
```

Set Filter

Sets a Filter value to all matching within an XDC. The Filter must be one of the "Available Filters".

Parameters:

XDC	required	Name/ID of XDC
Name	required	Name of Filter
Value	optional	Value(s)

Example:

```
Set Filter XDC="MyCache" Name="Customer State" Value="Texas;Florida"
```

Set Enabled/Disabled

This command allows you to turn on/off processing for an XDC.

Parameters:

XDC	required	Name/ID of XDC

Examples:

```
Set Disabled XDC="SalesData" "

Set Enabled XDC="SalesData"
```

Set CacheKey

This command allows you to set the key to be used when creating a multi-value Cache.

This will be used when an XDS uses either a Webi Document with prompts/filters or a DB Query with @placeholders.

Parameters:

XDC	required	Name/ID of XDC
Key	required	Name of unique param

Examples:

```
Set CacheKey XDC="SalesData" Key="State"
```

Execute(exec) XDC

This command submits a request to start an XDC Refresh. Based on the loading on the InfoBurst system the XDC may be queued.

The command will display the runtime ID that can be used to track the progress and/or display detailed actions. The variable "$RTSID" will be set to the runtime ID.

The "To" option will invoke a WAIT XDC and then an EMAIL RTLOG to allow you to run a burst, wait for it to finish and send the runtime log in one command.

Parameters:

Name	required	Name/ID of XDC
To	optional	Email address to send RTS Log
RunAs(user)	optional	InfoBurst user account to use
Password(pwd)	optional	InfoBurst user password

Example:

```
Exec XDC Name="SalesData"

Exec XDC 1234 to="jwilcox@infoburst.com"
```

Wait XDC

This command will wait for the last executed XDC to finish to allow a script to be written that could perform an action afterwards such as emailing the runtime log etc.

The default delay is "5" seconds.

Parameters:

Delay	optional	# seconds between checking status

Example:

```
Wait XDC

Wait XDC Delay="2"
```

Set XDCAlertMode

This command will set the XDC Alert Mode.

There are three possible modes:

- None No alert emails will be sent
- All An alert email will be sent for EVERY xdc abort
- Selected An alert email will be sent for SELECTED xdc aborts

To see the current mode, use the “list config” command and look at the value for “XDCAlertMode”.

It is also possible to set the alert mode using set config XDCAlertMode=”x” where x is either 0, 1 or 2 (0=None, 1=All, 2=Selected).

Parameters:

Mode	required	**XDCAlertMode**

Add Alert

This command will add a Burst to the list of **selected** XDS’s to monitor in case of an abort.

Parameters:

XDC	required	ID or Path\Name of XDC

Remove Alert

This command will remove a Burst from the list of **selected** XDS’s to monitor in case of an abort.

Parameters:

XDC	required	ID or Path\Name of XDC

Schedules & Events

These commands allow you to create/modify/delete Schedules and Events.

To view a Schedule or Event, use the standard display command (e.g. display schedule “mySchedule”)

To rename or delete a Schedule or Event, use the standard Rename/Delete commands.

To see all pending Schedules, use the “list schedule” command.

New Schedule / Replace Schedule

Create or Replace a Schedule. If Replace command used any existing object with the same name in the same Folder will be deleted first.

Type defaults to Schedule_Burst.

Frequency defaults to Once with a Start Date of Today and a Start Time of + 2 minutes.

Expires defaults to 6 (hours).

Enabled defaults to True.

Priority defaults to 0 (lowest)

If no folder is given, the current folder will be used.

If using the Days option, “M-F” can be used to specify Monday thru Friday, “SS” can be used to specify “Sat+Sun” and “*” or “All” means every day.

To set a time equal to right now, set Start=”now”.

Parameters:

Name	required	Name of Schedule
Description	optional	Description
Type	optional	**`ObjectSubType`**
Frequency(when)	optional	**`ScheduleFrequency`**
StartDate(date,starting)	optional	Starting Date
StartTime(start,at)	optional	Starting Time (HH:MM)
EndDate(until,thru)	optional	Ending Date

EndTime(ending)	optional	Ending Time (HH:MM)
Interval(every)	optional	#Mins/Hours/Days if EveryXX
Expires	optional	# Hours before Schedule expires
Enabled	optional	True/False
Days(weekdays,on)	optional	“Mon,Tue”, “1,2,10” or “1/1/09,2/2/09”
Priority	optional	0 thru 99
RunAs	optional	InfoBurst User account
RunAsPassword	optional	InfoBurst User account password
Bursts	optional	Initial list of Bursts to add “b1,b2”
Events	optional	Initial list of Events to add “evt1,evt2”
OnStart	optional	Initial list of Start Actions “ac1,ac2”
OnComplete	optional	Initial list of Complete Actions “ac1,ac2”
OnAbort	optional	Initial list of Abort Actions “ac1,ac2”
Delim	optional	Delimiter for Days/Bursts/Events “,”
Folder	optional	Folder to store list

Example:

```
New Schedule Name=”DailySales” when=”Daily” start=”08:00” on=”M-F”

New Schedule Name=”StockLevel” when=”everyminute” every=”15”
start=”08:00” ending=”17:00”
```

Modify Schedule

Modify an existing Schedule.

If using a name for a Schedule, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Schedule
NewName	optional	New name for the Object
Description	optional	Description
Frequency(when)	optional	**`ScheduleFrequency`**
StartDate(date,starting)	optional	Starting Date
StartTime(start,at)	optional	Starting Time (HH:MM)
EndDate(until,thru)	optional	Ending Date
EndTime(ending)	optional	Ending Time (HH:MM)
Interval(every)	optional	#Mins/Hours/Days if EveryXX
Expires	optional	# Hours before Schedule expires
Enabled	optional	True/False
Days(weekdays,on)	optional	“Mon,Tue”, “1,2,10” or “1/1/09,2/2/09”
Priority	optional	0 thru 99
Delim	optional	Delimiter for Days/Bursts/Events “,”
Bursts	optional	List of Bursts to use
Events	optional	List of Events to use
OnStart	optional	List of Start Actions to use
OnComplete	optional	List of Complete Actions to use
OnAbort	optional	List of Abort Actions

Example:

```
Modify Schedule Name="DailySales" start="07:30"
```

Delete Schedule

Delete an existing Schedule.

If using a name for a Schedule, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Schedule

Example:

```
Delete Schedule Name="DailySales"
```

Add Burst

Add a Burst to an existing Schedule.

Parameters:

Schedule	required	Name/ID of Schedule
ID	required	Name/ID of Burst

Example:

```
Add Burst Schedule="DailySales" id="MyBurst"
```

Remove Burst

Remove a Burst from an existing Schedule.

Parameters:

Schedule	required	Name/ID of Schedule
ID	required	Name/ID of Burst

Example:

```
Remove Burst Schedule="DailySales" id="MyBurst"
```

Execute(exec) Schedule

This command submits a request to start each Burst/XDS in the Schedule. Based on the loading on the InfoBurst system the Burst(s)/XDS(s) may be queued.

If you wish to run the Schedule using a different user context to the one defined in the Schedule you can specify an optional User and Password.

Parameters:

Name	required	Name/ID of Schedule
RunAs(user)	optional	InfoBurst user account to use
Password(pwd)	optional	InfoBurst user password

Example:

```
Exec Schedule Name="SalesReports"

Exec Schedule "DailySales" user="jim" password="jimspwd"
```

Hold Schedule

This command holds a schedule that is waiting to run.

You can either use an instance value (obtained using "list schedule select="n") or give a Schedule ID/Time.

Parameters:

Instance	optional	Unique instance for Schedule
ID	optional	ID of Schedule
Time(at)	optional	Start time (required if >1 instance)

Example:

```
Hold Schedule ID="100"
```

Release Schedule

This command releases a schedule that has been held.

You can either use an instance value (obtained using "list schedule select="n") or give a Schedule ID/Time.

Parameters:

Instance	optional	Unique instance for Schedule
ID	optional	ID of Schedule
Time(at)	optional	Start time (required if >1 instance)

Example:

```
Release Schedule ID="100"
```

Force Schedule

This command forces a schedule that is waiting to run to run now.

You can either use an instance value (obtained using "list schedule select="n") or give a Schedule ID/Time.

Parameters:

Instance	optional	Unique instance for Schedule
ID	optional	ID of Schedule
Time(at)	optional	Start time (required if >1 instance)

Example:

```
Force Schedule ID="100"
```

Stop Schedule

This command stops a schedule that is already running.

You can either use an instance value (obtained using “list schedule select=”n”) or give a Schedule ID/Time.

Parameters:

Instance	optional	Unique instance for Schedule
ID	optional	ID of Schedule
Time(at)	optional	Start time (required if >1 instance)

Example:

```
Stop Schedule ID=”100”
```

New Event / Replace Event

Create or Replace an Event. If Replace command used any existing object with the same name in the same Folder will be deleted first.

Type defaults to Event_Trigger.

AutoReset defaults to True.

CheckEvery defaults to “minute”.

CheckInterval defaults to “1”.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of List
Type	optional	**`ObjectSubType`**
Source	optional	Path to File or Path to DBConn.Qry
TrueValue(value)	optional	True value for Qry
ResetSource	optional	Path to DBConn.Qry for AutoReset
AutoReset	optional	“True” to reset when Event is set
CheckEvery(check)	optional	second/minute/hour
CheckInterval(every)	optional	# seconds/minutes/hours to check
Trace	optional	“True” to log each event check
Folder	optional	Folder to store Object

Example:

```
New Event Name=”DailyWHLoad” type=”file”
source=”\\rptserver\triggers\dwh.txt”

New Event Name=”SysReady” type=”query” source=”SQL.CheckSys”
check=”second” every=”30” resetsource=”SQL.ResetSys”
```

Modify Event

Modify an existing Event.

Parameters:

Name	required	Name/ID of Schedule
NewName	optional	New name for the Object
Type	optional	**`ObjectSubType`**
Source	optional	Path to File or Path to DBConn.Qry
TrueValue(value)	optional	True value for Qry
ResetSource	optional	Path to DBConn.Qry for AutoReset
AutoReset	optional	True to reset when Event is set
CheckEvery(check)	optional	second/minute/hour
CheckInterval(every)	optional	# seconds/minutes/hours to check
Trace	optional	"True" to log each event check

Example:

```
Modify Event Name="DailyWHLoad" check="second" every="15"
```

Delete Event

Delete an existing Event.

If using a name for an Event, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Event

Example:

```
Delete Event Name="DailyWHLoad"
```

Add Event

Add an Event to an existing Schedule.

Parameters:

Schedule	required	Name/ID of Schedule
ID	required	Name/ID of Event

Example:

```
Add Event Schedule="DailySales" id="DailyWHLoad"
```

Remove Event

Remove an Event from an existing Schedule.

Parameters:

Schedule	required	Name/ID of Schedule
ID	required	Name/ID of Event

Example:

```
Remove Event Schedule="DailySales" id="DailyWHLoad"
```

Actions

These commands allow you to create/modify/delete Actions.

To view an Action use the standard display command (e.g. display action “myAction”)

To rename or delete an Action use the standard Rename/Delete commands.

New Action / Replace Action

Create or Replace an Action. If Replace command used any existing object with the same name in the same Folder will be deleted first.

Type defaults to “settrigger”.

If no folder is given, the current folder will be used.

Parameters:

Name	required	Name of Schedule
Type	optional	**`ObjectSubType`**
Command(cmd)	optional	Command to run
Folder(dir)	optional	Working folder for Command
Event(trigger)	optional	ID or path for Event to trigger
Subject	optional	Email subject
Message(msg)	optional	Email message
To	optional	Email recipients
File	optional	File to Create
Contents	optional	Contents for File
Query	optional	Query to Run
Parameters(param,params)	optional	Parameters for Command / Query
Delim	optional	Delimiter if multiple To (“,”)
Folder	optional	Folder to store list

Recipients can be entered as follows:

- user@company.com
- [%macro]
- **user:**name
- **group:**name
- **list:**path\name or **list:**name
- **filelist:**path\name or **filelist:**name
- **ldaplist:**path\name or **ldaplist:**name
- **xilist:**path\name or **xilist:**name
- **mclist:**path\name or **mclist:**name
- **qry:**path\connection.qry or **qry:**connection.qry

A Column-name can be supplied (to over-ride any default) between ( ) and any conditions or query parameters can be supplied between { } for the following types:

- Filelist
- Ldaplist
- Mclist
- Qry

Example:

```
New Action Name="Trigger1" event="Wait1"

New Action Name="Alert1" type="email" subject="Daily sales"
message="has run OK" to="john@infoburst.com,steve@infoburst.com"
```

Modify Action

Modify an existing Action.

If using a name for the Action, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Schedule
NewName	optional	New name for the Object
Type	optional	**`ObjectSubType`**
Command(cmd)	optional	Command to run
Folder(dir)	optional	Working folder for Command
Event(trigger)	optional	ID or path for Event to trigger
Subject	optional	Email subject
Message(msg)	optional	Email message
To	optional	Email recipients
File	optional	File to Create
Contents	optional	Contents for File
Query	optional	Query to Run
Parameters(param,params)	optional	Parameters for Command / Query
Delim	optional	Delimiter if multiple To (“,”)

Example:

```
Modify Action Name=”Alert1” msg=”has completed”
```

Delete Action

Delete an existing Action.

If using a name for the Action, if no path is given then the path will default to the current folder.

Parameters:

Name	required	Name/ID of Action

Example:

```
Delete Action Name="Alert1"
```

Add Action

Add an Action to an existing Schedule, Burst or XDC.

Parameters:

Schedule	optional	Name/ID of Schedule
Burst	optional	Name/ID of Burst
XDC	optional	Name/ID of XDC
ID	required	Name/ID of Action
When	required	"start", "complete" or "abort"

Example:

```
Add Action Schedule="DailySales" id="EmailSalesTeam" when="complete"
```

Remove Action

Remove an Action from an existing Schedule, Burst or XDC.

Parameters:

Schedule	optional	Name/ID of Schedule
Burst	optional	Name/ID of Burst
XDC	optional	Name/ID of XDC
ID	required	Name/ID of Action
When	required	“start”, “complete” or “abort”

Example:

```
Remove Action Schedule=”DailySales” id=”EmailSalesTeam”
when=”complete”
```

Parameter Alias

These commands allow you to create/modify/delete parameter alias values that make it easier to work with database queries that operate on documents with differing prompt values.

Register ParameterAlias(Alias)

Create a new Parameter Alias.

Parameters:

Name	required	Name of Alias
Values	required	Delimited list of values
Delim	optional	Delimiter to use (default is “,”)

Example:

```
Register Alias Name=”State” values=”States,Enter one or more States”

Register Alias Name=”City” delim=”;” values=”Cities;Enter City Names”
```

Modify ParameterAlias(Alias)

Modify an existing Alias.

Parameters:

Name	required	Name/ID of List
Add	optional	Delimited list of Items to add
Remove	optional	Delimited list of Items to remove
Delim	optional	Delimiter to use (default is “,”)

Example:

```
Modify Alias Name=”States” Add=”NY,NJ”

Modify Alias Name=”States” Remove=”CA”
```

Unregister ParameterAlias(Alias)

Delete an existing Alias.

Parameters:

Name	required	Name/ID of List

Example:

```
Unregister Alias Name="States"
```

Network Paths

These commands allow you to register or remove UNC Paths with the system.

Register NetworkPath(Path)

Register a new UNC Path to be used during delivery or by a FileList object.

Parameters:

Location	required	UNC Path to register
ShortName(Alias)	optional	ShortName/Alias

Example:

```
Register Path location="\\reports\sales"

Register Path location=\\reports\sales\2008 ShortName="08Sales"
```

Unregister NetworkPath(Path)

Delete an existing UNC Path. The location will no longer be valid if used by a list or during delivery.

Parameters:

Location	required	UNC Path to unregister

Example:

```
Unregister Path location="\\reports\sales"
```

Network Printers

These commands allow you to register or remove Printers with the system.

Register NetworkPrinter(Printer)

Register a new Printer to be used during delivery.

Parameters:

Name	required	Name of Printer to register

Example:

```
Register Printer name="\\HP_Printer"
```

Unregister NetworkPrinter(Printer)

Delete an existing Printer. The location will no longer be valid if used during delivery.

Parameters:

Name	required	Name of Printer to unregister

Example:

```
Unregister Printer name="\\HP_Printer"
```

Object MetaTags

These commands allow you to register or remove MetaTags to help locate Objects within the repository.

Register Metatag(Tag)

Register a new Tag.

The Name must be between 1 and 10 characters and contain no spaces.

Parameters:

Name	required	Name of Tag
Description	optional	Description

Example:

```
Register MetaTag name="08Sales" description="Sales Reports 2008"
```

Unregister MetaTag(Tag)

Delete an existing Tag.

Parameters:

Name	required	Name of Tag

Example:

```
Unregister MetaTag name="08Sales"
```

Set Metatag(Tag)

Assign a Tag to an Object. To assign more than one Tag, use a "," delimiter.

Parameters:

ObjectID	required	ID of Object
Tags	required	Tag(s)

Example:

```
Set MetaTag ObjectID="100" Tags="08Sales"
```

System Configuration

These commands allow you to change system settings.

Use the “list config” command to see all current values.

Set Config

Set one or more system values.

Example:

```
Set Config MaxBursts=”15”

Set Config EmailServer=”msexchange1” EmailServerPort=”25”
```

Appendix A – Values for special types

Where a comma separated list is shown, you can use either value.

BurstAlertMode

- **None** No alerts
- **All** Alert on ALL Burst aborts
- **Selected** Alert on SELECTED Burst aborts

BurstDestination

- **Network,Net,NetShare** Network Share
- **Email** Email
- **FTP** FTP
- **XI_Folder** BusinessObjects XI Folder
- **XI_InBox** BusinessObjects XI User InBox
- **XI_Instance** BusinessObjects XI Instance
- **WebDav** WebDav
- **Printer** Printer
- **Custom** Custom delivery using a command-line
- **RoamBI** Mellmo RoamBI ES

BurstFormat

- **Native,rpt,rep** Original Format
- **Pdf** PDF Format
- **Xls,Excel** Excel Format
- **Htm,Html** HTML Format
- **Text,Txt** Text Format
- **Csv** CSV Format
- **Delim** Delimited Text Format
- **Xml** Xml Format
- **Png** Image Format

BurstParameterSource

- **Inline** Hard-coded value or values
- **Qry,DBQry,Query** Use Database Query
- **MultiColumnList,MCList** Use MultiColumn List
- **List** Use List
- **FileList** Use FileList
- **LdapList** Use LdapList
- **XIList** Use XIList
- **Report,rpl** Use Report Element

DBConnectionType

- **SQL, SQLServer** Microsoft SQL Server 2000+
- **ORA, Oracle** Oracle
- **Access,MSAccess** Microsoft Access
- **ODBC** Generic ODBC

DocumentProcessingMode

- **ServerOnDemand,OnDemand** Use Platform "Report Engine"
- **ServerInstance,Instance** Use Platform Scheduler
- **ClientOnDemand,Client** Use Client (Desktop Intelligence only)

DBQryUsage

One or more of the following values delimited by ","

- **EPL** Query will be used to supply a single document parameter/filter
- **EDL** Query will be used to supply a list of email recipients
- **EVT** Query will be used to test for an event
- **XDS** Query will be used as an Xml Data Source
- **DOC** Query will be used to supply one or more document parameters/filters

PlatformType

- **XI20_Server,XIR2** BusinessObjects XIR2 Enterprise Platform
- **XI25_Server_XIR25** BusinessObjects XIR2 with Service pack 5
- **XI26_Server_XIR25** BusinessObjects XIR2 with Service pack 6
- **DeskI_XIR2** BusinessObjects XIR2 Client (Desktop Intelligence)
- **XI3_Server,XI3** BusinessObjects XI3.0 Enterprise Platform
- **DeskI_XI3** BusinessObjects XI3.0 Client (Desktop Intelligence)
- **XI31_Server,XI31** BusinessObjects XI3.1 Enterprise Platform
- **XI32_Server,XI32** BusinessObjects XI3.1 SP2 Enterprise Platform
- **XI33_Server,XI33** BusinessObjects XI3.1 SP3 Enterprise Platform
- **MS20_Server,MS20** Microsoft Reporting Services 2005/2008

PlatformAppServerTechnology

- **.net,dotnet** Microsoft .Net
- **Java** Java

PlatformAuthType

- **E, Enterprise** Default authentication for Platform
- **A, AD,ActiveDirectory** Use Active Directory authentication
- **W, Win,Windows** Use Windows authentication
- **L, Ldap** Use LDAP authentication
- **SAP** Use SAP authentication
- **IB,InfoBurst** Use InfoBurst User Pass-Thru

PlatformConnectionType

- **Default** Default value
- **W,ws,webservice** Use Web Services
- **L,localclient,client** Use local connection

LdapSearchScope

- **Base** Search from the base
- **1level,onelevel** Search 1 level deep
- **Subtree** Search all sub-levels

ListColumnType

- **String** String
- **Number,Numeric** Number
- **Date** Date

ObjectSubType(Action)

- **Command,Cmd** Action is to run a Command.
- **Email,SendEmail** Action is to send an Email
- **Trigger,SetTrigger** Action is to set a Trigger Event
- **CreateFile** Action is to Create a new File
- **RunQuery** Action is to run a Query
- **StartBurst** Action is to start a Burst

ObjectSubType(Event)

- **File** Event is based on a File.
- **Query,Qry** Event is based on a Database Query
- **Trigger,Bool** Event is based on a Boolean Trigger

ObjectSubType(Schedule)

- **Burst** Schedule contains one or more Bursts
- **XDS** Schedule contains one or more Xml Data Sources

RTSDeliveryMode

- **Immed,Inline** Deliver after each Document
- **End,AtEnd** Deliver at end of Burst

RTSGenerationMode

- **Seq,Sequential** Generate one by one
- **Par,Parallel** Generate in parallel

ScheduleFrequency

- **Once** Schedule occurs Once
- **Daily** Schedule occurs Daily
- **DaysInMonth** Schedule occurs on specific days in the month
- **EveryMin,min,minute** Schedule occurs every so many minutes
- **EveryHour,hour** Schedule occurs every so many hours
- **EveryNNDays,everyday** Schedule occurs every so many days
- **Manual** Schedule must be started manually
- **Event** Schedule is not based on any date/time but event(s)
- **OnTheseDates,Dates** Schedule occurs on the given dates
- **Macro** Schedule occurs on dates based on Macros

UserCacheContent

- **File** Files
- **DBQryExec** Query Results

UserRole

- **Admin, Sysadmin** System Administrator
- **Oper, SysOper** System Operator
- **Manager, ProjectManager** Manager
- **User, TeamMember** User
- **Consumer, ReportConsumer** Consumer
- **Recipient, ReportRecipient** Recipient

XDCType

- **Webi** BusinessObjects Webi Document
- **DbQry,Query,Qry** Database Query
- **Excel** Excel workbook

XDSColumnTotals

- **None** No Totals
- **Top** Totals on Top
- **Bottom** Totals on Bottom

XDSRowTotals

- **None** No Totals
- **Left** Totals on Left
- **Right** Totals on Right

XDSTotalType

- **Sum** Sum of values
- **Min** Minimum value
- **Max** Maximum value
- **Avg** Average of values
- **Wta** Weighted Average of values

XDSNumberFormat

- **None** No formatting (default)
- **Int** Integer
- **Div100** Integer divided by 100
- **Div1000** Integer divided by 1000

Appendix B - Using `ObjectFilter` to restrict which objects to list

When listing objects in a folder or using the more general Find command, you can specify a list of object types and for Documents, sub-types.

If you do not specify anything for Objects, all objects will be listed.

For example, to list the contents of the "\Sales" folder but only list documents

Objects="Document"

To list Documents and Bursts

Objects="Document,Burst"

The available object types are

- "Action"
- "Document" or "Doc"
- "Burst"
- "DBConnection"
- "LdapConnection"
- "Text"
- "List"
- "FileList"
- "LdapList"
- "XIList"
- "MultiColumnList" or "MCList"
- "Schedule"
- "Event"
- "XDC"

To restrict a Document to one or more types, enclose the list of document types within brackets () with a delimiter of either '|' or ':'

Document(CR|WI) or Document(CR:WI)

The following document sub-types are supported

- "WI" BusinessObjects WebI Documents
- "DI" BusinessObjec ts DeskI Documents
- "CR" Crystal Reports
- "BO" BusinessObjects WebI and DeskI Documents
- "PDF" Adobe PDF Documents
- "DOC" Microsoft Word Documents
- "XLS" Microsoft Excel Documents
- "PPT" Microsoft PowerPoint Documents
- "TXT" Text Documents
- "SWF" Adobe Flash Documents
- "XC" Crystal Xcelsius Documents

When listing documents on a Platform, the available object types are

- "WI" BusinessObjects WebI Documents
- "DI" BusinessObjec ts DeskI Documents
- "CR" Crystal Reports
- "BO" BusinessObjects WebI and DeskI Documents
- "PDF" Adobe PDF Documents
- "DOC" Microsoft Word Documents
- "XLS" Microsoft Excel Documents
- "PPT" Microsoft PowerPoint Documents
- "TXT" Text Documents
- "SWF" Adobe Flash or Crystal Xcelsius Documents